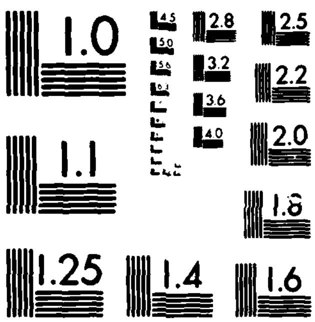
AD-A163 027 AN EXPERT SYSTEM DECISION AID FOR A COMMAND CONTROL AND COMMUNICATIONS OPERATOR(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D L TOBAT 1/4

UNCLASSIFIED DEC 85 AFIT/GE/ENG/85D-50 F/G 17/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AN EXPERT SYSTEM DECISION AID FOR A COMMAND, CONTROL AND COMMUNICATIONS OPERATOR

THESIS

Daniel L. Tobat
Captain, USAF

AFIT/GE/ENG/85D-50

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

AFIT/GE/ENG/85D-50

AN EXPERT SYSTEM DECISION AID FOR A COMMAND, CONTROL AND COMMUNICATIONS OPERATOR

THESIS

Daniel L. Tobat
Captain, USAF

AFIT/GE/ENG/85D-50

AFIT/GE/ENG/85D-50

AN EXPERT SYSTEM DECISION AID FOR A COMMAND, CONTROL AND COMMUNICATIONS OPERATOR

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Accesion For		
NTIS CRA&I	☑	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

by

Daniel L. Tobat, B.S.E.E.

Captain, USAF

December 1985

Approved for public release; distribution unlimited

Preface

The purpose of this study was to demonstrate the feasibility of using an expert system to build a decision aid for a C3 operator. The completed system actually performs the simulated tasks of the strategic missile warning officer and was turned over to the Aerospace Medical Research Laboratory for further development and testing. This project demonstrates an architecture and explanation capability that is applicable to many expert systems, that can and should assist human operators by letting the machine do what it does best.

I have a great personal interest in the field of expert systems, and although they are not the panacea to every current computational problem, they will be making a significant technological impact in the years to come.

I wish to express my thanks to Captain Steven K. Rogers, my advisor, for his encouragement and advice in this project and Captain Stephen E. Cross for his instruction in the field of Artificial Intelligence. I would also like to thank Mr. Don Monk of the Aerospace Medical Research Laboratory for providing me the support and facilities to properly perform this work, and Mr. Kurt Mayrand of the Systems Research Laboratory for his invaluable assistance on the Xerox 1108. Finally, I wish to thank my wife Susan for her support and understanding through the nine long months of this project.

Daniel L. Tobat

Table of Contents

Table of Contents

List of Figures

List of Figures

List of Tables

Index to SENTINEL Functions and Rule Sets

Index to SENTINEL Functions and Rule Sets

Abstract

The growing complexity and quantity of information used in Command, Control and Communications (C3) networks makes it essential to reduce the workload on the operators of these networks. The SENTINEL project uses the Artificial Intelligence concept of an expert system to produce a decision aid for the strategic Missile Warning Officer, using a simulation of a C3 network that involves multiple missile launches and up to 20 countries. In this research, a blackboard model expert system using rule bases and object oriented programming techniques was developed to permit SENTINEL to offer several layers of explanation. SENTINEL analyzes patterns and causes of reported events into higher level yet less precise forms to provide the upper layer of explanation. SENTINEL deals with uncertainty by using the statistical concepts of feature sets and decision thresholding. The feature sets represent the essential characteristics of a launch event and are evaluated to see how well they fit a particular hypothesis. The decision threshold used to select an interpretation is determined by appraising the distance from each hypothesis, as well as by previous events. This project demonstrates the feasibility of building expert system decision aids for C3 operators by using specialized explanation capabilities, and reasoning with uncertainty in a more statistically conventional way.

AN EXPERT SYSTEM DECISION AID FOR A COMMAND, CONTROL AND COMMUNICATIONS OPERATOR

I. Introduction

At the heart of many of today's military systems is a command, control and communications (C3) network. For such diverse systems as the Airborne Warning and Control System (AWACS), the North American Aerospace Defense Command (NORAD) and some of the systems proposed for the Strategic Defense Initiative, a vast flood of information about enemy activity on the ground, in the air and throughout the electromagnetic spectrum will be collected, transmitted, analyzed and acted upon through C3 networks. Just as the C3 networks permit quicker and more effective responses, the consequences of mistakes and failures in the systems and their operators have multiplied.

Although information processing and data automation are heavily used to collect, transmit and display C3 data, the key decisions and analysis of the data are in the hands of a few highly trained operators. These operators are still subject to human frailties, and few attempts have been made to help them in arriving at decisions. This situation is quite understandable given the difficulty of decision making and the need to keep weapons under tight control. In view of the growing complexity and quantity of the information used in C3 networks, it becomes essential to improve the command and control of military forces by aiding the performance of the C3 operator.

The field of Artificial Intelligence by the use of an "expert system," offers the opportunity to build a decison aid which would assist the C3 operator in performing his task. Expert systems are computer programs that are designed to represent and apply specialized knowledge in a narrow field of expertise (14:3). Such a decision aid could increase warning times and improve control of military forces during a highly dynamic situation where humans would be fatigued and begin making mistakes.

Background

Rather than building a general decision aid for any C3 network, a specific C3 operator task and environment was selected. The C3 network used as a model for this study is the Simulated C3 Operator Performance Evaluation (SIMCOPE) developed by the Harry G. Armstrong Aerospace Medical Research Laboratory (AAMRL) at Wright-Patterson AFB, Ohio. SIMCOPE was completed in February 1984 and models the task of a Strategic Missile Warning Officer (MWO) in a fictional scenario. Basically the operator must acknowledge and appropriately respond to messages concerning sensor status, reported launch indications and intelligence conditions. Based upon this information and his own training, the SIMCOPE Missile Warning Officer must decide whether transpiring events represent a hostile or test event, and subjectively assess the probability of his judgements being correct. SIMCOPE has been used by the AAMRL to test how subjects respond to different display formats and to varying stress levels in performing tasks similar to the ones an actual MWO would have to

make. The SIMCOPE simlulation has been viewed by several members of NORAD who felt it was fairly realistic (11).

Numerous expert system computer programs have been built since the mid 1970's in such diverse fields as medical diagnosing, computer configuration and geological prospecting. Expert systems employ up to thousands of IF-THEN rules to symbolically process information and make inferences, and are characterized by their capability of explaining their reasoning process and being able to deal with uncertainty. The task of the expert system builder is to construct a program that can represent and reason with the knowledge of a domain expert. These programs take a considerable amount of time to develop and require extensive and continous validation and modification. See Appendix A for a further description of expert systems.

Problem

Building a decision aid for SIMCOPE involves trying to solve two distinct problems: to build an aid that helps the operator overcome inherent human bias and also to overcome the limitations of conventional computer programs which do not easily provide the capability of a decision aid. The SIMCOPE operator, like any other human operator is subject to various biases in decision making which can cloud his perception and result in an improper analysis and judgement of the situation at hand. Andrew P. Sage enumerates these biases into 25 separate categories. Among the most applicable for a C3 operator are the following:

A). Adjustment and Anchoring: a difficulty that arises out of excessive data and information. To reduce the mental effort required, the operator selects a particular chain of thought as an anchor, and then attempts to incorporate the rest of the data into the same mold, resulting in a flawed analysis.

B). Availability: The operator uses only readily available information and ignores other data. An event is believed to be highly probable if it is easy for the operator to recall similar episodes.

C). Data Saturation: The operator reaches a premature conclusion on too small a sample of information and will stop the acquisition of further data prematurely.

D). Conservatism: the failure to revise estimates based on receipt of new contradictory information. This is also related to the data saturation bias.

E). Desire for Self-Fulfilling Prophecies: This is a form of selective perception where the operator values a certain interpretation or outcome and only analyzes the information which supports this conclusion.

F). Ease of Recall: Events that can be recalled easily are given more weight in decision making than events that are difficult to recall.

G). Expectations: Operators will attach higher validity to information that confirms their previous beliefs. The presence of large amounts of information makes it easier to selectively ignore nonconfirming evidence and thus prove anything one wants to prove (19:647).

The presence of stress or fatigue may of course substantially increase the probability of these biases ocurring. Having established that C3 operators are indeed prone to biases which could adversely affect their judgements in this important task, one may wonder why conventional programming methods are not used within the context of the SIMCOPE simulation to assist the operator. The answer to that question is to be found in the SIMCOPE Instruction Manual in the section entitled "Recommendations for Making Judgements" the manual instructs aspiring SIMCOPE operators that:

> These judgements are left for you to decide either because the computer cannot be instructed or expected to make such decisions. The computer cannot be instructed if the process for making the judgement is incompletely described or includes factors that the a machine will not know. The computer cannot be expected to make judgements about intention or motivation . . . The judgement of calling an act hostile or a threat is not something one wants to entrust to a blind, mechanical process. The consequences of error in this case are too serious to allow a fully automated decision process (1:75).

By using the techniques of Artificial Intelligence, the objections based above on conventional programming methods can be overcome to provide an objective, helpful, but not fully automated decision process. While many of the allegations above are true for conventional methods of computer programming, part of the reason for the increasing popularity and use of expert systems is that they do provide the ability to make decisions and handle uncertainty by symbolically processing information. The central problem is to build an objective decision aid that can assist the human operator by not falling prey to subjective biases and can make judgements and deal with uncertainty in a logical way.

Objectives

The objective of this project will be to build a decision aid that can assist the C3 operator in making judgements by being an independent "bench mark" to insure that factors have not been forgotten or that subjective human bias has not interfered with the decisions. By being able to recall all available information, the decision aid will produce the same result each time given the same inputs, and will not be prone to fatigue, stress or subjective bias. The objective here is not to replace the human operator. Instead, the goal is to implement the principal task of computer aided support which has been described as "augmenting human capabilities in need of augmentation, while not diminishing abilities in those areas in which human capabilities exceed those of the computer" (19:645). A true decison aid permits the human operator to accept or disregard its advice and will assist him most during peak periods when the MWO is more prone to making mistakes in recalling events and in analyzing the situation.

The objective of the SIMCOPE decision aid, henceforth referred to as SENTINEL, will be to prove the concept of a decision aid for the MWO rather than provide an operational model that is ready to be fielded. Before SENTINEL is fielded an extensive verification program would have to be established and the system would need to be optimized for speed and accuracy. Minimizing the processing time is essential, since a decision aid will be most needed during hectic situations when intelligence and sensor reports will arrive with increasing frequency. However, for the purpose of this project it is

necessary to show only that SENTINEL is computationally feasible to implement.

The characteristics of SENTINEL are that of a event driven decision aid that uses a blackboard model and works by forward chaining. The decision to use a forward chaining rule based architecture for SENTINEL was influenced by other expert systems that perform on-line real time interpretation of multiple sensors monitoring a machine process (12:158). SENTINEL must also be able to deal with the uncertainty inherent in a process where knowledge of enemy forces and intentions is incomplete. It must also be able to explain to the C3 operator how it arrived at a conclusion by showing what factors were considered in reaching that outcome. This process will hopefully remind the operator of certain factors that may have been overlooked. An explanation capability is of great importance in getting operators to feel confident about the system and to help them recognize the limitations of SENTINEL.

In spite of the increasing number and type of expert systems in use, building one is a very time consuming task. Several factors made it possible to complete SENTINEL in the time alloted for this effort. Instead of trying to develop an expert system that does the job of an actual Missile Warning Officer, SIMCOPE has already conveniently quantified and somewhat abstracted the job of the MWO. Also, the use of expert systems building tool software that aids in the building of rule bases was used to cut down on the development time required.

Scope

This study will only use the SIMSCOPE simulation of a missile warning officer's tasks as a model for the decision aid. Other types of C3 operator's tasks will not be considered, although much of this study is applicable to other C3 tasks as well. Certainly the C3 operators on an AWACS during a pitched aerial battle or the operators for the some of the systems associated with the proposed Strategic Defense Initiative could benefit from a similar decision aid. A full fledged expert system that functions as a decision aid for the MWO in the SIMCOPE simulation is developed. This expert system will not be optimized for speed or accuracy.

Assumptions

This study will be subject to the limitations of the SIMSCOPE scenario as to how well it models the task of an actual MWO. SIMSCOPE's main limitation is that only a fixed number of possible intelligence and sensor reports can be generated. This is a valid assumption if the intelligence and sensor data is computer processed before it reaches the strategic missile warning officer. SIMCOPE also has a physical limitation in that it portrays a 3 dimensional problem, the one of assessing the threat posed by incoming ballistic missiles in a 2 dimensional space both on its display screens and in the information provided by satellite reports. The only information provided by the satellites in SIMCOPE concerning the missile trajectory are the launch coordinates and a heading. More information on SIMCOPE messages can be found in Appendix B.

Materials and Equipment

This project was developed in the Command, Control and Communications Operator Performance Engineering (COPE) Facility, under the auspices of the Human Engineering Division of the Harry G. Armstrong Aerospace Medical Research Laboratory (AAMRL) at Wright-Patterson AFB, Ohio. The hardware used was a Xerox 1108 "Dandelion" LISP machine with 40 megabytes of hard disk storage space. The machine software used was Xerox's "Harmony" version of the INTERLISP-D language. The "Buttress" release of the LOOPS Expert System Building Tool, also developed by Xerox, was used to construct the rule bases used in this project. Further information about the use of LOOPS can be found in Appendix C. The SIMCOPE simulation was developed by several members of the Human Engineering Division of the AAMRL during 1983. SIMCOPE runs on a PDP-11 and was programmed in FORTRAN. The only existing documentation of the SIMCOPE simulation is the "SIMCOPE Instructions Manual" which was written by the AlphaScience corporation in February, 1984 (1).

Approach

The approach used to develop SENTINEL differs greatly from the conventional process used to build expert systems. In the conventional expert system building process, the "knowledge engineer" interviews a "domain expert" in an effort to capture his specialized knowledge into IF-THEN rules. In the case of SIMCOPE, no experts actually exist since it is only a simulation. However, since SIMCOPE is an evaluation tool for the MWO task it presents an ideal

starting point for demonstrating the feasibility of a decision aid for the MWO. Also, some researchers such as Gary Klein, object to the technique of using domain experts as a source of rule based knowledge.

> We cannot trust the verbal reports of subject matter experts on how they perform tasks. They will tell us about the rule-following portion of their behavoir. Our questions about what they are doing often seem to be attempts to discover the rules they are following, so they try to imagine plausible rules to satisfy us. There is no reason to suspect that an expert will come up with better rule-governed explanations for behavior than an engineer, or than a psychologist (15:301).

The approach used for capturing the rule based knowledge used in SENTINEL was to use the "judgement" and "operating procedures" sections from the SIMCOPE instruction manual as the initial "expert." These procedures or heuristics clearly delineated the difference between threatening, hostile events and test events, but were not very useful for events that fell between these extremes. To deal with these "gray" events, SENTINEL's knowledge was gradually modified using common sense and facts of the scenario in a manner that was consistent with prior SIMCOPE procedures. Thus, SENTINEL was first built to handle the obvious, clearly separated cases, and then cautiously and slowly extended by incorporating almost any imaginable SIMCOPE event.

SENTINEL also differs from conventional expert systems in the way it interacts with the user. In many expert systems the user enters into a dialogue with the system which poses questions to which the user responds in an almost "natural" language. Because

SENTINEL is built around SIMCOPE, it operates the way the simulation operator does; by obtaining information about launch events from messages provided by SIMCOPE rather than from a dialogue with the operator, and by providing answers to the formatted reports which the SIMCOPE operators respond to.

In view of the departures from the conventional expert system framework, it is necessary to address the question "Is SENTINEL really an expert system?" SENTINEL is an expert system because it captures the specialized knowledge of the SIMCOPE operator in the narrow domain of the strategic missile warning task. SENTINEL also has many of the qualities that distinguish expert systems: it reasons with uncertainty, deals with search, can explain its reasoning and is capable of revising its conclusions in the presence of conflicting information. These hallmarks along with the use and development of rule bases and objects constitute evidence that SENTINEL is indeed an expert system. In view of the tremendous amount of change and advances in the field of expert systems as well as Artificial Intelligence in general, it is necessary to quickly summarize the current state of technology in respect to how it applies to SENTINEL.

Summary of Current Knowledge

Expert system computer programs have been in use for many years in such diverse tasks as mineral prospecting, (PROSPECTOR: Duda, 1979) diagnosing bacterial infections (MYCIN: Shortliffe, 1976) and configuring computers (R1: McDermott, 1982). The first half of the

decade of the 1980's has witnessed a virtual explosion in the number and type of expert systems described in the general Artificial Intelligence literature. In spite of the vast number of existing expert systems, and although the need for expert systems in C3 is well recognized (13:1), the application of a decision aid for a C3 operator that is used for prediction and strategic military forecasting appears to be unique. Two systems that are similar to SENTINEL is some respects are an expert system for indications and warning analysis developed by ESL Incorporated and AIRPLAN, which features decision making in time critical situations.

The Indications and Warning Analysis (IWA) expert system assists a military intelligence analyst by assimilating incoming intelligence reports and predicting where and when an armed conflict may erupt next. It utilizes two separate blackboards along with three levels of abstraction: specific, higher level and very general states. An interesting feature of the IWA expert system is that it can run both forward in time for prediction and back in time to fill in past gaps in its knowledge when new information becomes available (16:260). SENTINEL differs from this system in that its intelligence input is much more limited and deals with a time period of minutes instead of weeks. Running backwards could be beneficial for SENTINEL as a type of computer learning, but that has very little direct impact on the task of aiding the MWO.

The AIRPLAN system is concerned with managing the launch and recovery of aircraft aboard a carrier, which entails making decisions in a time critical situation. It does this by updating its

display when a report is received and then determines the available options which are characterized as good or bad. If AIRPLAN does not have a backlog of unanalyzed reports, it continues its analysis to come up with a set of recommendations (17:234). SENTINEL does a similar sequence of processing reports, but only for the purpose of interpreting data rather than developing plans. However, the most significant difference between the two systems is that the SENTINEL task entails a great deal of reasoning with uncertainty. To understand this portion of SENTINEL's operation it is necessary to examine how current expert systems deal with uncertainty.

One widely used approach to handling uncertainty in rule based systems such as MYCIN has been the use of certainty factors. As described by Patrick H. Winston, certainty factors are numbers between 1 and 0 that are intended to reflect how certain a fact is, with 0 indicating a definitely false fact and 1 a definitely true fact. These certainty factors are propagated across rules in inference networks through various methods to derive an output certainty factor. Some of the simplest procedures that combine the certainty factors of various antecedent facts into a deduced fact are taking the certainty of the weakest or strongest argument supporting it, or else multiplying the output by an "attenuation factor" (23:188). For example; in a rule of the form:

If A and B and C THEN D,

where A has a certainty factor of .6, B one of .5 and C one of .3 could result in a certainty factor of .6 by taking the strongest argument or .3 by taking the weakest. More elaborate methods for

using certainty factors exist but most of them have little or no support from elementary probability theory. In order to deal with uncertainty in a manner consistent with probablity theory, various Bayesian methods of representing and combining uncertainty information for expert systems have been developed. One of the most well known Bayesian methods of computing confidences in rule based systems is the Dempster-Schafer theory which permits the assignment of probability mass to general propositions constructed from other propositions (8:9).

Difficulties exist with these two main schools of thought in reasoning with uncertainty. The problem with certainty factors is that with a single value one has no information about its precision, and when evidence for and against a proposition is combined into a single number, one cannot distinguish between disbelief and lack of evidence. The difficulty with Bayesian methods is that use of Bayes theorem requires masses of data to determine what type of statistical relationship exists between all the factors involved. In almost all current systems, subjective judgements are used in place of this data which render their results prone to the risks of inaccuracy and inconsistency (6:18). In the case of SENTINEL, one does not wish to use the subjectiveness of certainty factors to simply replace human biases with statistical bias. As for Bayesian methods, the nature of the factors used in SENTINEL do not permit either strict statistical independence or dependence to be shown. For example, in cases where intelligence reports indicate an impending attack which are then corroborated by other indications,

the intelligence reports are statistically dependent. However, the case where intelligence does not indicate an attack but other factors suggest an attack, implies statistical independence. Clearly, SENTINEL must use a different way of dealing with uncertainty and must be a fairly unique expert system because no other expert system has been built before for this task.

Overview

For purposes of analysis, SENTINEL can be divided into the five major blocks pictured in figure 1. The METASCHEDULER receives all SIMCOPE messages and routes them accordingly. The SCENARIO REPRESENTATION block gives SENTINEL the same information which is already available to the MWO when he looks at the SIMCOPE graphical display, and is triggered upon receipt of the first launch indication, which is a satellite report labeled ADS-1. The First Pass Messages include the processing of SYS system status messages, INT intelligence reports as well as the completion of ADS-1 processing. The Second Pass Satellite Report represents the processing of the second launch indication, which is labeled as ADS-2. The Third Pass Radar Report represents the processing performed upon receipt of the third and final launch indication. Note that the events themselves are propagated separately from the SIMCOPE message routing.

The following discussion of the SENTINEL decision aid largely follows the chronological sequence of its development. The next chapter will describe the METASCHEDULER and SCENARIO REPRESENTATION

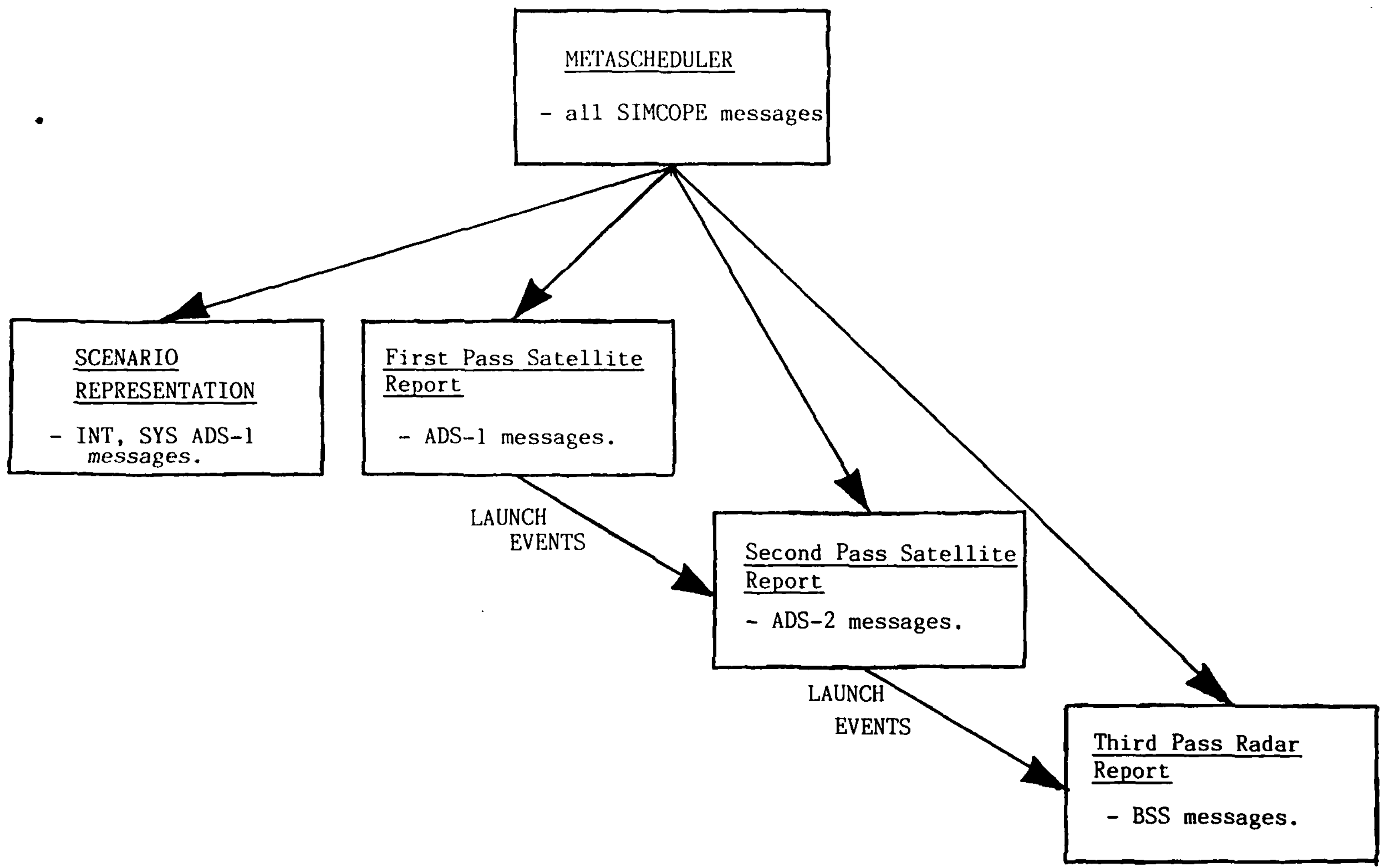


Figure 1. General Block Diagram of SENTINEL.

blocks, while Chapter III will discuss the First Pass Message processing. Chapter IV will cover the Second Pass case and will describe how dealing with uncertainty is accomplished. Chapter V will cover the Third Pass and the conclusions and recommendations. Appendix A, "A Basic Description of Expert Systems," is recommended for the reader unfamiliar with knowledge based systems. Appendix B, "Outline of SIMCOPE Operations," is background material for those readers who do not have access to the SIMCOPE instruction manual. Appendix C, "Evaluation and Use of LOOPS," illustrates how the LOOPS expert system building tool software operates and describes its limitations. Finally, Appendix D "Miscellaneous Functions and Program Listings," covers SENTINEL features not discussed elsewhere and presents all the documented rule bases and LISP functions which make up SENTINEL.

II. METASCHEDULER and SCENARIO REPRESENTATION

Format

To standardize the format used for the remainder of this discussion, the following rules will apply. The major blocks which describe a combination of functions and rule sets will consist of all capital letters such as METASCHEDULER. The actual functions, rule sets and LISP variables which comprise SENTINEL are written as they appear in the LISP programming for SENTINEL. Therefore, LISP functions, property lists and variables all appear in capital letters such as SENSORNUMBER or SCHEDULER. The rule sets are a combination of capital and uncapitalized letters such as BoardVoting or MIXEDConfidence. The blackboards, or objects used in SENTINEL all end in "Board" such as INTELBoard or DETECTBoard. The actors in SENTINEL will also be fully capitalized such as LAUNCHSITE or TARGET. The first major block, METASCHEDULER will demonstrate the use of functions, actors, rule sets and LISP variables.

METASCHEDULER

The METASCHEDULER block consists of a function by the same name and a rule set labeled MSGHandler. The input to METASCHEDULER varies from all the other functions in SENTINEL by being any SIMCOPE message. METASCHEDULER's task is to set the common parts of each incoming SIMCOPE message to a variable. For example, a typical SIMCOPE ADS-1 report is in this form:

```
((CWC (ADS ((S1 142913) 6) -30) 14 (143410) 9))
```

The CWC indicates the destination of the message, in this case the Command Warning Center, the ADS reveals this is a satellite report and the S1 indicates this is a first pass message from the southern ADS satellite. METASCHEDULER then would set CWC to be the LISP variable named DESTINATION and S1 to be the SENSORNUMBER. All the SIMCOPE messages share the same format for the first three elements, but the structure of the rest of the messages varies considerably. The only modification to the actual form of the SIMCOPE messages as they appear to the operator is the substitution of parentheses for slashes to make the LISP manipulation of this form easier. Further details about the structure of each type of message and what all the elements are can be found in Appendix B.

Once METASCHEDULER has set the common variables, MSGHandler actually routes the messages thorugh SENTINEL. Using such variables as SENSORNUMBER, MSGHandler uses 7 rules to provide an output to the user and direct the SIMCOPE message to the appropriate function for processing. For example, if an incorrect destination is received, MSGHandler advises the user of this ocurrence and asks if the operator wishes to go ahead and process the message anyway. Before going on to discuss the details of how this processing is achieved, one must understand how the SCENARIO REPRESENTATION captures the information presented to the Missile Warning Officer since this determines the structre of almost all the other functions in SENTINEL.

SCENARIO REPRESENTATION

The objective of the SCENARIO REPRESENTATION of SENTINEL is to accurately convey to the expert system the same type of information the MWO receives by looking at the SIMCOPE display, which is shown in figure 2. Depicted on a geographic grid, lies the country of THEM on the right hand side of the map, and on the left the country of US. All of the known or suspected launch sites in the country of THEM are represented by triangles or a Star of David which are explained in the table underneath the map. The possible targets in the county of US are represented by large or small dots with alphabetic labels such as "RVN" by the major cities. The solid circles located on the East coast of US and numbered 1 through 7 are the radar coverage provided by Barrier Site Surveillance (BSS) sites. The two large dotted circles which cover the country of THEM represent the satellite coverage provided by Aerospace Defense Satellites (ADS) South and North. The two types of reports provided by these satellites and by the BSS site are most of the "events" which are communicated to the MWO.

In addition to receiving and responding to the ADS reports, the MWO receives a great deal of information from the visual depiction of a launch event. Figure 3 shows what can be seen shortly after receiving the second report on the same launch event (referred to as an ADS-2 report) from one of the two satellites. The dotted lines indicate the wedge shaped "fan" which represents the detected heading of the missile along with a possible variation of plus or minus 5 degrees. One of the tasks of the MWO is to mentally extend

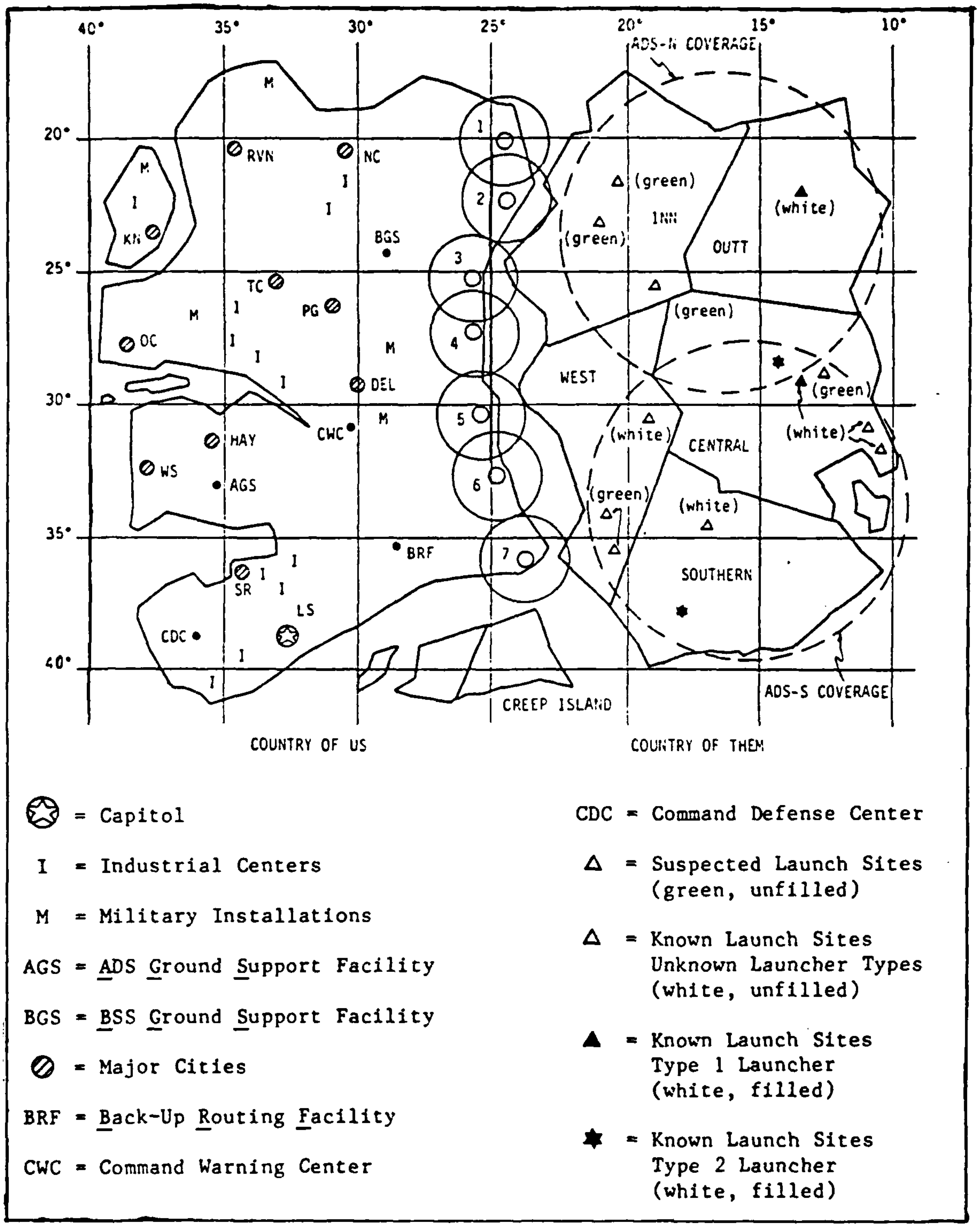
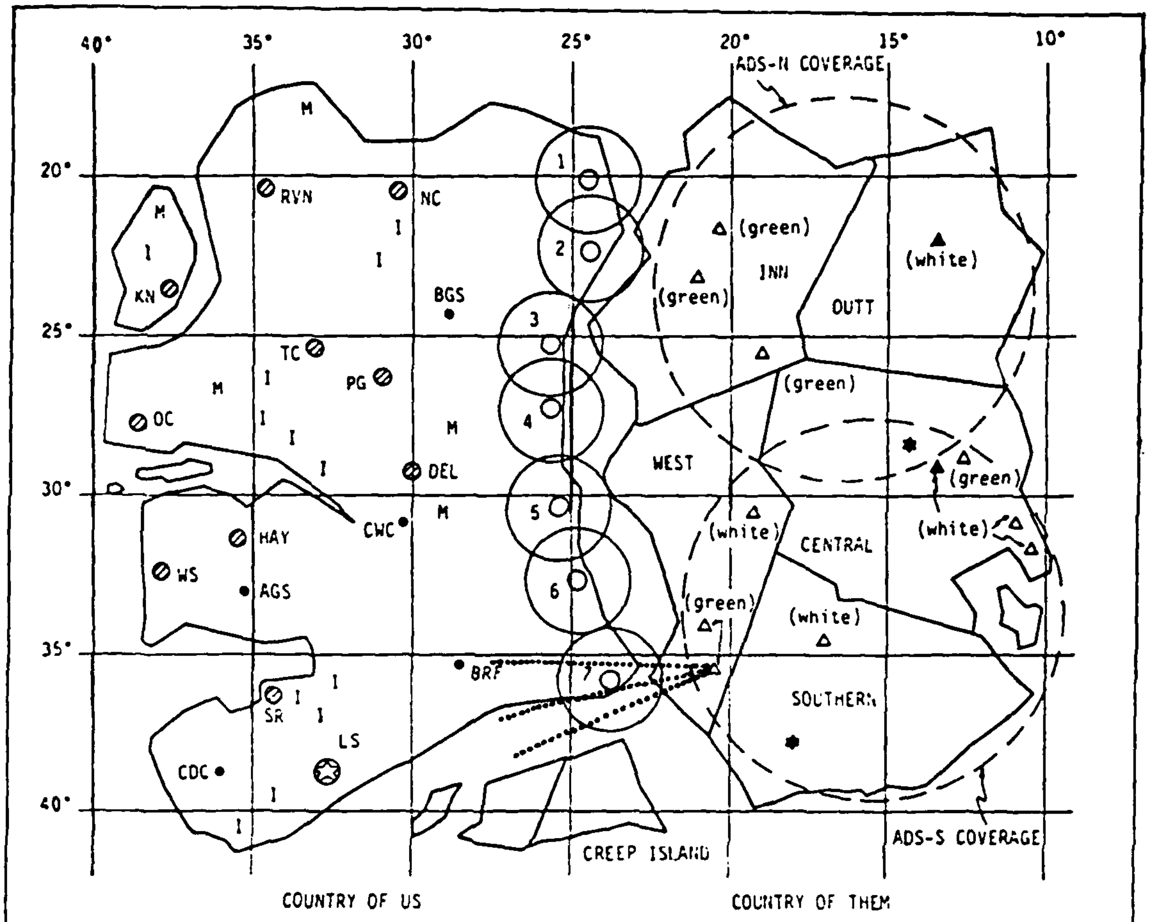


Figure 2. The SIMCOPE Detail Map (1:3).



✪ = Capitol		CDC = Command Defense Center	
I = Industrial Centers		△ = Suspected Launch Sites (green, unfilled)	
M = Military Installations		△ = Known Launch Sites Unknown Launcher Types (white, unfilled)	
AGS = ADS Ground Support Facility		▲ = Known Launch Sites Type 1 Launcher (white, filled)	
BGS = BSS Ground Support Facility		✸ = Known Launch Sites Type 2 Launcher (white, filled)	
⊘ = Major Cities			
BRF = Back-Up Routing Facility			
CWC = Command Warning Center			

Figure 3. The SIMCOPE Map Showing a Missile Launch (1:3).
(Detected Missile Heading is the Center Line.)

this wedge into the country of US and predict which BSS site will be able to detect the missile and to predict which targets are threatened. In figure 3 for example, it appears that BSS site #7 at the extreme southern end of US is likely to detect the incoming missile, and that the cluster of targets surrounding the capital in the southwestern peninsula of US could be struck. Armed with this information, the MWO must be able to predict whether a hostile attack or a test launch is underway and provide a subjective assessment of his degree of confidence in this prediction.

There are two other types of reports produced by SIMCOPE which do not deal directly with a launch event. The INT intelligence reports may indicate that the country of THEM is believed to be planning a test launch or preparing an attack. The SYS system reports indicate any outage problems due to sabotage or failure at a sensor, and when it might be expected to regain full capability. SYS and INT reports do not follow a general sequence as the launch events do; however the majority of the reports deal with launch events.

The objective of the SCENARIO REPRESENTATION is to permit SENTINEL to capture and represent the information available to the MWO, given the map and access to the ADS reports. SENTINEL must be able to also perform the tasks the MWO does visually by predicting which BSS site will detect the launch event and what the possible targets of the launch event are. Both SENTINEL and the MWO can profit from knowing that the launch reports follow a specified pattern. The sequence of events explained below tracks a single

missile launching from the initial to the final report which is shown to the MWO.

Sequence of Events

The usual sequence of events following a missile launch in the SIMCOPE simulation consist of receiving two distinct satellite reports and a BSS (radar) report. The first indication of a launch event will be the ADS-1 satellite report, which yields the latitude and longitude coordinates of the launch event and the reported missile infrared intensity. This intensity reading can be sufficient in some instances to determine what type of missile has been launched. Missiles fall into two categories: TYPE1 missiles have a short range and are armed with a single warhead with limited maneuvering capabilities, while the TYPE2 corresponds to a longer ranged, multiple warhead missile whose warheads can maneuver freely. The second indication of a launch event will be the confirmation of the first report by the ADS-2 satellite report which yields further information about the missile such as its heading. Finally, a BSS report will be sent by the BSS site that detects the missile which further confirms the type and heading of the missile. In order to automate SIMCOPE it is necessary to give SENTINEL the ability to determine which targets are threatened by the launch and which BSS site will detect the missile. Since the type of information available to SENTINEL consists of geographic coordinates and a heading, this strongly suggests a polar type of representation.

Justification for a Polar Representation

As shown in figure 4 a polar representation consists of a set of vectors originating from the launchsite which are labeled by a range and a heading to various targets. This representation is an efficient way of generating headings to different targets and comparing them to the reported heading of the missile. Knowing the range and headings to various targets enables SENTINEL to determine what specific target the missile could strike. For example, suppose a TYPE1 missile has been detected with a heading of 160 degrees whose maximum range is 800 kilometers. As shown in figure 4, target A with a heading of 153 degrees lies closest to the reported heading, but it can be ruled out since the distance to target A exceeds the maximum range of the missile. Target B on the other hand is within the range of the missile though its heading deviates further. Target C can be ruled out completely because of its extreme deviation from the reported missile heading though it lies well within the range of the TYPE1 missile. In order to compute a polar representation, the latitude and longitude from each target and launch site needs to be accessed rapidly when needed. The basic mechanism for storing this information will be a property list which is discussed in the following section.

The Actors

Actors are the sensors, launch sites and targets that are the subject of SIMCOPE. Information for each of these three types of actors can be stored in a "property list" under the particular name

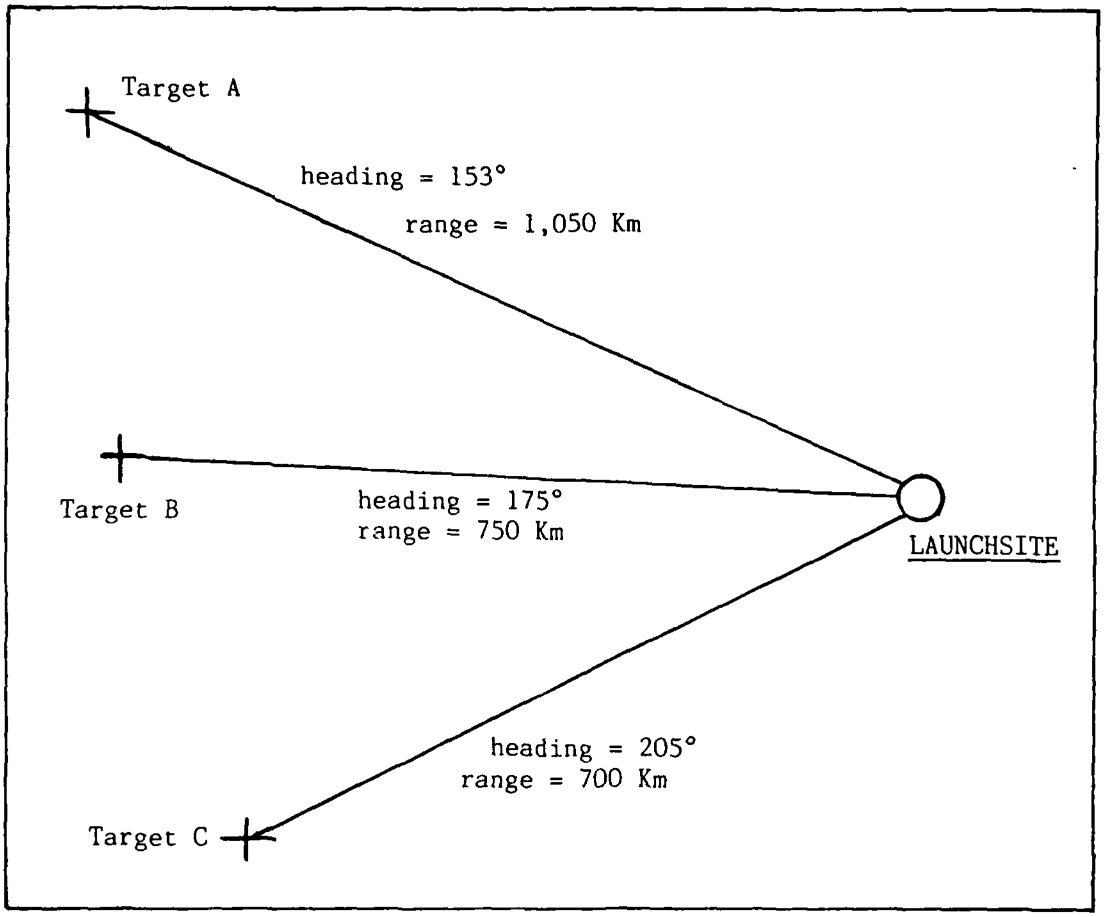


Figure 4. Polar Representation of Three Targets From One Launchsite.

of a site such as TGT1. The PUTPROP and GETPROP LISP commands permit the effective retrieval of this data at any time. The object oriented capabilities of LOOPS were not used for the representation of the actors because of the inability of LOOPS to utilize variables as an address, as described in Appendix C. Because LOOPS objects would not have permitted multiple unknown launchsite actors, property lists which did not have this limitation were used.

The simplest type of actor is the TARGETs. Each TARGET is represented by a property list with the four attributes shown in figure 5. The latitude and longitude measurements come from the detail map and are accurate to within half a degree. A listing of all the sites and their attributes can be found in Appendix B. The TYPE label identifies the target as a city, a C3 node or a military or industrial facility. There are a total of 33 targets in the SIMCOPE scenario, and they are grouped into five different lists. The high value target list is composed of the capital of US and the C3 warning network, while the other four lists group targets regionally in horizontal bands from north to south. For example, target area one covers all the targets to the north of 24 degrees of latitude and which are covered by BSS sites one and two.

The SENSORs are represented by a property list with the six attributes shown in figure 5. The labels which differ from the TARGET's property list are the RANGE and the CAPABILITY. Assuming circular radar coverage, the RANGE represents the maximum radius of the BSS site's radar. The identical size of the circles found in the Detail Map represent a range of approximately 185 kilometers. Making

Property List of TARGET

Name:
Latitude:
Longitude:
Type: (Military, Industrial, City, C3)

Property List of SENSOR

Name:
Latitude:
Longitude:
Type: (BSS, ADS North or South)
Capability: (Full, Partial or Zero)
Range: (in kilometers)

Property List of LAUNCHSITE

Name:
Latitude:
Longitude:
Type: (known or suspected, Type1 or Type2)
HiValTag: (for high value targets)
BSSTag: (for BSS site detection)
FanCall: (the rest of the targets)

Figure 5. Property List of the Three Actors in SENTINEL.

the range of coverage a variable rather than a constant permits equipment malfunctions or local weather variations to affect the radar. However, the area of coverage is still assumed to be circular. Finally, the CAPABILITY label permits the capability of the BSS or Satellite sensor to be identified as FULL, PARTIAL or ZERO as indicated by the latest SYS system report. The CAPABILITY status of the SENSOR will bear heavily on the reliability of the information reported by that sensor.

The LAUNCHSITEs are the most complex type of actor in SENTINEL because they have seven attributes and they are the key to the remainder of SENTINEL processing. Besides the usual labels, LAUNCHSITEs also have a TYPE label which corresponds to whether they are a known or a suspected TYPE1 or TYPE2 launch site as previously seen as a table on Figure 1. The BSSTAG contains the maximum and minimum headings that a missile could take from a particular launch site and still be detected by a particular BSS radar site. The HIVALTAG represents the headings a missile would have to take to strike the capital of US or a C3 node. Finally, FANCALL consists of the polar representation to all the other (non high value or BSS site) targets. The polar representations for FANTAG, BSSTAG and HIVALTAG could certainly be computed beforehand for a multitude of launch sites and stored explicitly. However, this "brute force" approach has definite drawbacks if the number of targets is increased to a few hundred and if the missile launchers are mobile and can fire from almost any location. Therefore, it is appropriate to be able to compute the polar representation for a previously

unknown launch site when the ADS-1 report is received. This is the objective of the LISP functions described below.

Block Diagram of the SCENARIO REPRESENTATION

As shown in figure 6, five functions comprise the SCENARIO REPRESENTATION and the function which supports all this processing is POLAR. Given a set of launch and target coordinates in terms of latitude and longitude, POLAR returns the range and heading from the launch site to the target. Given a list of targets, FAN calls on POLAR repeatedly to compose an association list, whose elements consist of the target name, heading and range. HIVALCALL uses FAN to compute the range and heading to all the high value targets. The high value target list consists of the capital of US and the C3 nodes of the strategic warning network of which the MWO is a part. BSSCALL uses POLAR directly to compute the maximum and minimum headings from a launchsite that could be detected by a specific BSS site. Finally FAN is used to compute the range and heading from the launch site to the rest of the 33 targets. SCHEDULER invokes all the other functions upon receipt of an ADS-1 report of a launch event. Thus, the polar representation is only computed when a launch indication is received, which permits a launch emanating from any location to be processed, rather than just being limited to sites whose geographic coordinates are known. Having covered the overall SCENARIO REPRESENTATION the following is a detailed discussion of each separate function.

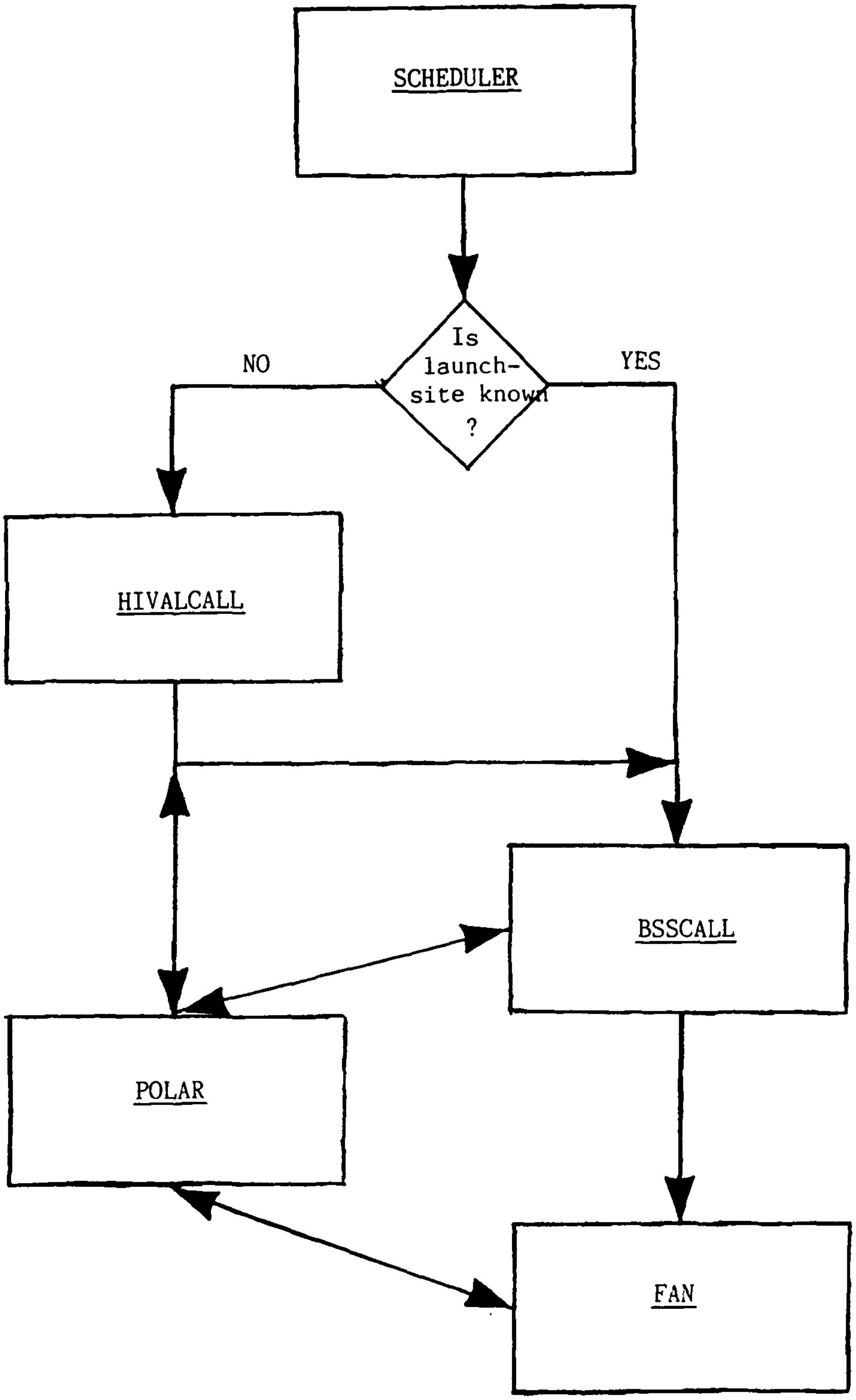


Figure 6. Block Diagram of the SCENARIO REPRESENTATION Functions.

POLAR

The POLAR function enables the computation of a polar representation. Its inputs are the launch coordinates (latitude, longitude) and target coordinates (latitude, longitude). It is invoked by using (POLAR '(23 34) '(30 20)) where the first pair indicates that the target has a latitude of 23 degrees, and the second pair is the coordinates of the launch site. As shown in figure 7, POLAR uses trigonometric principles to solve for the arctan of the angle (heading) and the hypoteneuse (range). Because the targets could lie in any direction away from the launch site, it is necessary to correct the angle to a geographic heading by using the correction factors in each quadrant shown in figure 7. The special cases of 90, 180, 270 and 360 degrees need to be handled separately because either the divisor or the dividend will be zero which can represent either 180 or 360 degrees. Therefore, it is handled as a special case. Since the hypoteneuse or range will originally be expressed in degrees, it is necessary to convert the distance to kilometers by making each degree of longitude equal to 111 kilometers. Because the length of a degree of longitude varies as one moves towards the polar regions, this is the standard approximation which is valid at the equator. This limitation is common to maps which attempt to represent a three dimensional object in only two dimensions. Although POLAR can compute a polar representation for one site, it is usually necessary to compute the representation for an entire list of targets, which is what FAN does.

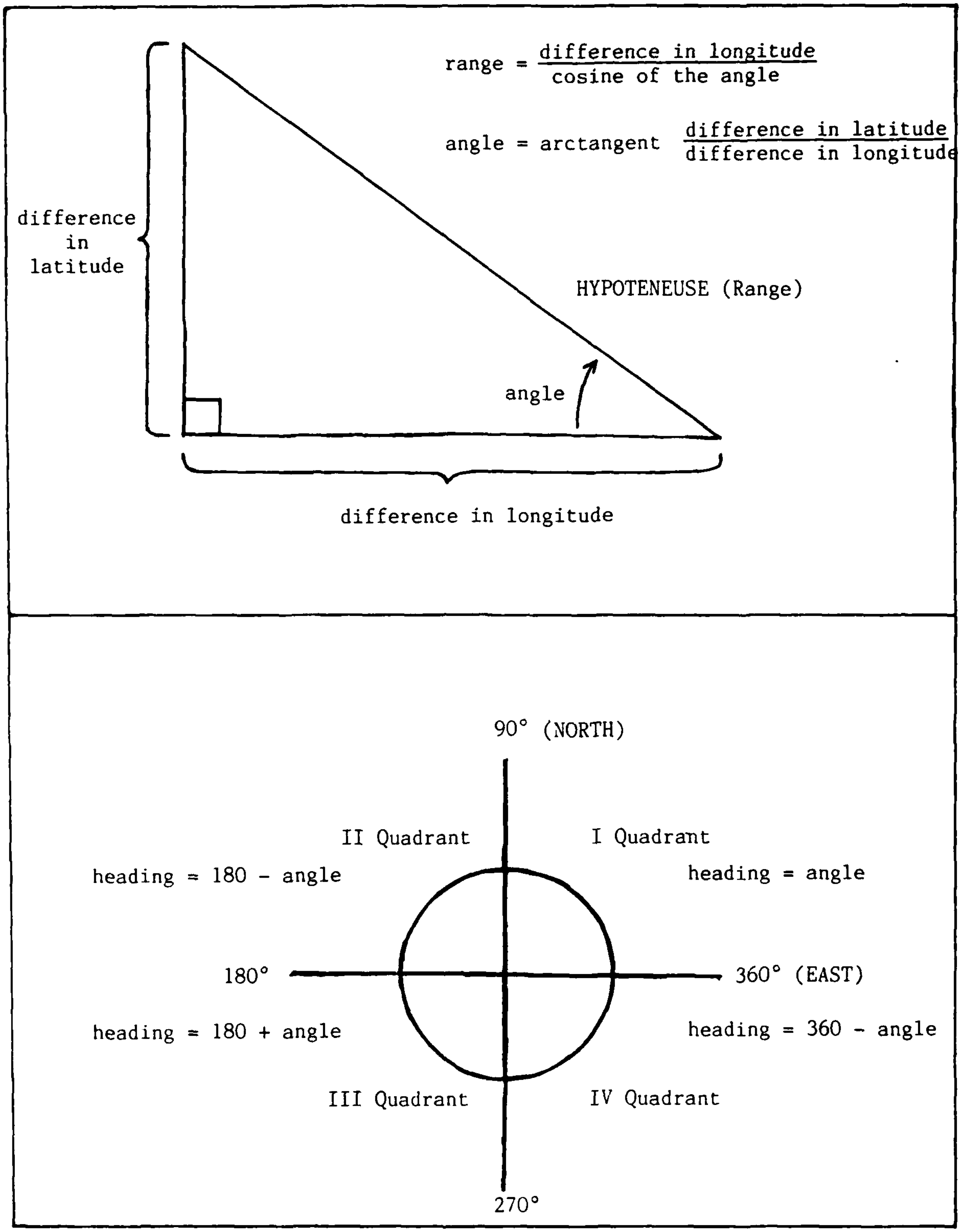


Figure 7. Detail of POLAR's Computations and Heading Correction Factors.

FAN and HIVALCALL

Given a list of targets and a launch site, FAN calls on POLAR for each individual target, and produces an association list of the computed range and headings. The target lists that might be given to FAN are the High Value Target List and the four Target Area Lists. Values for the latitude and longitude of the targets and the launch site are retrieved from the property lists of these actors. The list produced by FAN has a label for its first element and looks like this:

```
((TARGET  HEADING   RANGE)
 (TGT1     345      1047)
 (TGT2     215      1756) ... and so on.)
```

FAN is invoked by typing in (FAN 'LAUNCHSITE), and it will compute the range and heading from the given launchsite to all the targets in the list TARGETLIST. Therefore, it is necessary to reset TARGETLIST to the appropriate list each time. FAN is invoked by both HIVALCALL and SCHEDULER. HIVALCALL sets the TARGETLIST to the high value target list, calls on FAN and embeds the resulting association list on the HIVALTAG label on the property list of the LAUNCHSITE actor. Note that HIVALCALL is the only polar representation that is previously computed and stored for all the known or suspected LAUNCHSITEs. HIVALCALL determines the HIVALTAG of that LAUNCHSITE actor's property list, only when the launch site is unknown or upon initialization of the decision aid. FAN is also invoked directly by SCHEDULER to complete the polar representation from the launchsite to all the remaining (non high value) targets. The result is stored

under the FANCALL label of the LAUNCHSITE's property list.

BSSCALL

One of the challenges in representing the information for SENTINEL is capturing the circular radar coverage presented by a BSS site in a polar representation. To effectively use the polar representation, it was necessary to represent by just a few points the circular area of coverage provided by radar. The first attempts to represent the BSS coverage were by the "two point" methods, shown at the top of figure 8. The top circle represents a BSS site covered by the inscribing the largest size rectangle inside the circle that determines a northeast and southeast corner that would cover most of the headings that could be detected by that site. The bottom circle represents using a fixed angle technique to come up with the corners. The problem with these techniques is shown by the missile track with an especially "sharp" heading of 120 degrees that could easily penetrate both BSS sites without falling into the range of headings that would be detected by either of the "two point" methods.

A better approach is to use four points as shown at the bottom of figure 8. Given a range and a center point for the BSS site, the latitude and longitude of the four points representing the furthest reach of the circle along all four directions is computed. BSSCALL is invoked by (BSSCALL 'LAUNCHSITE), and given the property lists of the LAUNCHSITE and the BSS sites (to include RANGE!), BSSCALL uses

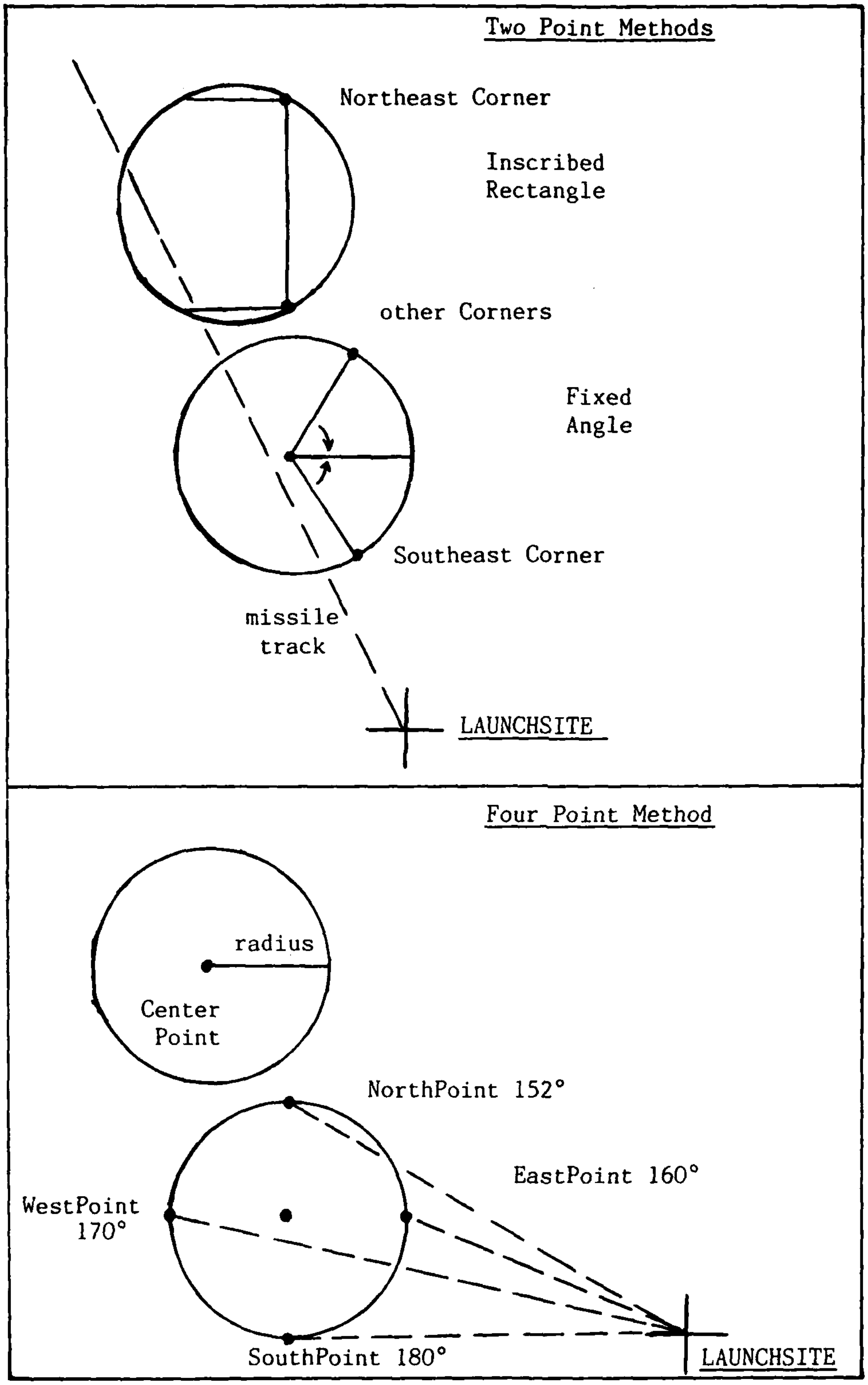


Figure 8. Two and Four Point Method for Computing BSS Detection in BSSCALL.

POLAR directly to compute the heading to all four points on the BSS circle. Originally, BSSCALL called on FAN but this proved cumbersome since the range information is redundant and the TARGETLIST needed to be reset four times. The output of BSSCALL at this point looks like this:

```
((LAUNCHSITE   NORTH  SOUTH  EAST  WEST)
   (BSS#1       152    180    170   160)
   (BSS#2       143    165    158   150)... and so on through BSS#7)
```

It becomes apparent that only the maximum and minimum headings need be stored without losing any information. In the above example, BSS site#1 could detect any missile from the given launch site that has a heading between 152 degrees (the minimum) and 180 degrees (the maximum). Thus the output of BSSCALL is simplified to:

```
((LAUNCHSITE  MAX   MIN)
   (BSS#1      180   152)  (BSS#2 165 143) ... and so on.)
```

After this association list is completed, it is attached to the LAUNCHSITE actor under the BSSTAG label. Having computed the polar representation for all the targets and BSS sites, it is now necessary to describe how these functions are invoked.

SCHEDULER

The input to SCHEDULER differs from the other functions by being the "raw" ADS-1 report generated by SIMCOPE. SCHEDULER first picks out the reported latitude and longitude of the launch event. For example, if ADS-1 is:

```
((CWC (ADS ((S1 142718) 8) -29) +13) (142915) 8))
```

Then a launch was detected at 29 degrees of latitude and 13 degrees of longitude. The first thing SCHEDULER does is to compare the reported latitude and longitude to the limits of that satellite's region of coverage. Thus if the southern ADS coverage is normally limited to latitudes of 30 degrees or greater, then SCHEDULER would announce this fact to the operator and permit him to stop the processing of this report. This feature permits the operator to continue processing if the satellite coverage region deviates from the normal or to stop if he suspects the report is erroneous.

Having insured that the reporting satellite is normally in a position to record this particular launch event, SCHEDULER then compares the reported coordinates to those of all the known launch sites. If a match is made, SCHEDULER announces the match, retrieves the previously computed HIVALTAG from the property list of the LAUNCHSITE, and then calls on BSSCALL and FAN to compute additional association lists and embeds them on the property list of the LAUNCHSITE. If no match is found, SCHEDULER creates a new property list for the previously unknown launch site and computes the HIVALTAG as well by calling on HIVALCALL and then repeats the steps for the known launch site case.

After having processed thorugh the SCENARIO REPRESENTATION, one can now examine the property list of the LAUNCHSITE actor. Therefore, for the previously unknown launch site number 15 the response to the command (GETPROPLIST 'LS15) would be:

LABEL	VALUE	COMMENT
(LONG	23.5	(* Reported Longitude)
LAT	18	(* Reported Latitude)
HIVALTAG	((LS15 HEADING RANGE) (TGT1 143 1110) (TGT2 212 1027) and so on through TGT6)	
BSSTAG	((LS15 MAXPT MINPT) (BSS#1 134 145) (BSS#2 140 165) and so on through BSS#7)	
FANCALL	((LS15 HEADING RANGE) (TGT7 109 1369) (TGT8 205 2088) and so on through TGT33)	

One may now consider the computational feasibility of this technique. Use of the INTERLISP function TIME enabled measurements as to how much processing time was required to fully compute the polar representation for a launch site. For the current implementation of SENTINEL and given a previously undetected launch site, a maximum of 43.6 seconds of processing time was required for SCHEDULER to call HIVALCALL, BSSCALL and FAN for all 33 targets. Since in this case there is a very limited number of targets, iterating through all the targets is still largely a linear growth function in terms of computation time rather than an exponential growth one. Although, this technique was not used in SENTINEL the following is proposed as a way of limiting the computation of the polar representation for SENTINEL.

Reduction of Computation Time

Once the BSSTAG has been computed and a reported heading for the missile has been detected, the following procedure can reduce the amount of computation time by only processing those targets which are likely to be attacked into a polar representation. This

takes advantage of the previous classification of all the non high value targets into geographic areas, as seen in figure 9. Heuristics can help reduce the computation time by limiting the number of targets that need to be expressed into polar form. For example, it appears obvious that if a missile is fired towards the northern portion of US as in launch site B from figure 9, it would be a waste of time to compute the polar representation for targets in the southern portion of the US that could not possibly be the objective of that particular missile. In the case illustrated at the top, launchsite A lies to the north of the detecting BSS site, and polar representation is limited to regions 3 and the regions to its north. Finally for the "head on" case where the missile heading is 180 degrees, only the targets in the area behind the detecting BSS site need to be computed into polar form. This technique of limiting the number of targets to be expressed into polar form would be useful for "real world" problems which could require hundreds of targets and thus trigger an exponential growth in computation time that could make this representation unfeasible.

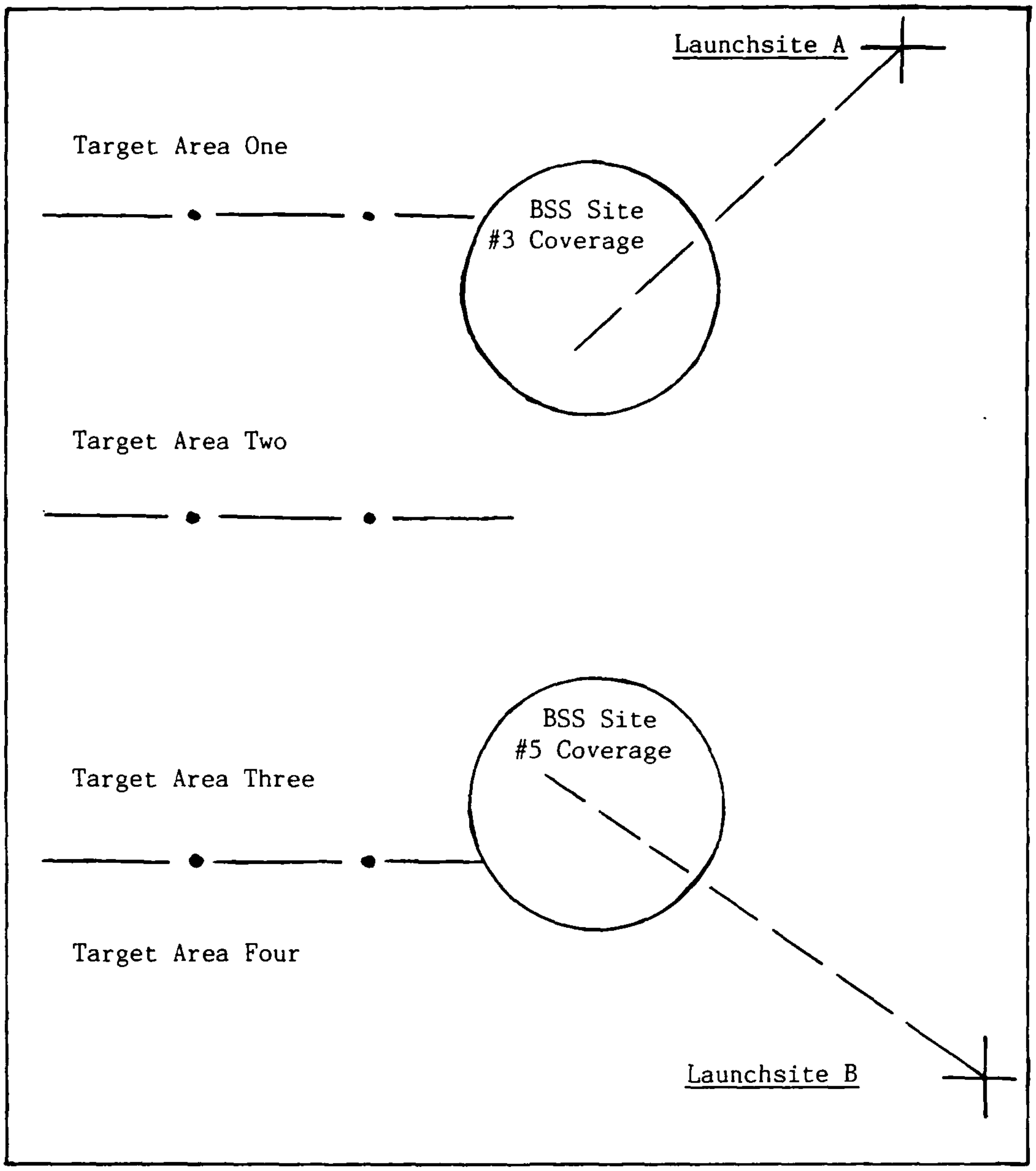


Figure 9. Limiting Target Areas Through the Use of FAN.

III. Processing of the First Pass Messages

There are five types of messages in the SIMCOPE simulation: the INT intelligence report, the SYS system status report, the ADS-1 first satellite pass report, the ADS-2 second pass satellite report and the BSS radar site report. The first pass messages which are covered in this chapter are the INT, SYS and ADS-1 messages. The simplest type of first pass message is the INT report.

Intelligence Reports

The INT reports in SIMCOPE cover the 20 fictional countries of the SIMCOPE scenario and inform the MWO if a given country has attacked, formed an alliance with or threatened an attack on another country. The reports also indicate which country originated the report, its source and the time and date at which the action can be expected to take place. Further details on this or any of the other messages can be found in Appendix B. INT reports differ from the other messages in the SIMCOPE scenario because they do not directly affect any actors. In contrast, SYS reports originate from a sensor actor, while ADS reports correspond to a launch site actor. Although INT reports are important, it is not obvious how to place the information they contain on a property list. The complexity of interactions of 20 countries made it necessary to simplify the analysis of INT reports by using a model of the SIMCOPE world.

Intelligence Model

ALthough 20 countries are used in the SIMCOPE simulation, it is somewhat misleading since no information is available on any countries except US and THEM. Indeed, the SIMCOPE map shows just those two countries, and it does not seem overly simplistic to only concentrate on reports from or about US and THEM. The situation that now results can be modeled as follows:

SOURCE reports from SOURCECOUNTRY indicate that REPORTEDCOUNTRY will or will not (INTELTYPE) do ACTIONZ, resulting in INTELCONDITION.

SOURCE can be OPEN indicating that it is publicly known information or INTEL indicating it was provided by an intelligence source of some kind. SOURCECOUNTRY should usually be US, since THEM is not likely to provide US with much information, and there is no way to ascertain the reliability of information from the other 18 countries. A similar problem exists with REPORTEDCOUNTRY; if it is one of the other 18 countries there is no information on how that could affect the MWO monitoring of THEM. Taking advantage of the discrete nature of SIMCOPE INT messages, ACTIONZ can be abstracted to result in the four following INTELCONDITION states:

TEST: a country has announced a test launch
RED: where a country has actually invaded or attacked
YELLOW: where a country has threatened invasion or attack
GREEN: which reflects moves towards greater stability such as reports of surrenders or alliances which would tend to dampen the possiblilities of further conflict.

Since TEST would entail presumably high confidence levels of a launch indication without being a hostile event it is treated as a

separate category. INTELTYPE reflects whether the INT report is of a positive type such as "has surrendered" versus the negative type as in "has not surrendered." Heuristics were used to represent this situation by using a rule that states that all negative type reports result in a YELLOW condition except for negative YELLOW. In that case, represented by "has not threatened attack" a resulting condition of GREEN is more appropriate. INTELCONDITION is what the SIMCOPE labels as Readiness States, and summarizes the overall state of the SIMOCPE world. The information needed to represent the information contained in an INT message consists of the follwing:

INTELCONDITION		[RED YELLOW GREEN or TEST]
INTELTYPE		[Positive or Negative]
SOURCE		[OPEN or INTEL]
REPORTEDCOUNTRY	SOURCECOUNTRY	[One of the 20 SIMCOPE countries]

Note that there is no need to represent the ACTIONZ since that information is captured in the INTELCONDITION. Only the SOURCE and the INTELCONDITION variables of the INT report will be used further because of the limitation on the MWO's knowledge of the other 18 countries, however the INT report must be shown to the user and the information above needs to be stored in a form accessible to all other SENTINEL functions.

Intelligence Output and Information Storage

In SIMCOPE no action is expected of the MWO when an INT report is received. All that is shown is the collection of numbers and slashes which correspond to the INT report. SENTINEL does translate

the message for the MWO but does not require an acknowledgement from the MWO. The output shown to the MWO from a typical INT report is as follows:

INTEL sources from the country of OVERTHERE reveal POSITIVE
indications of a GREEN condition for the country of: US

Having displayed all the five variables of the intelligence model to the user the question still remains of how to store this data. INT reports should not be attached to the property list of a single actor since they affect all of the actors but placing them on all of the actors would be awkward. The solution is to use a "blackboard" which is a method used in some expert systems that permits procedures to communicate with each other by posting messages and results on the blackboard (23:140).

In this case, the blackboard created was a LOOPS object entitled INTELBoard which has the five variables of the intelligence model stored as instance variables. LOOPS objects basically store information in a way that is almost identical to a property list and are discussed further in Appendix C. The first implementation of INTELBoard resulted in the blackboard being updated each and every time an INT report was recieved. The fallacy of this approach is that an INT report resulting in a RED condition could be written over and the value of the variables lost by a INT report about US that results in a GREEN condition. Therefore, the INTELBoard is updated only if the incoming INT report results in a more critical readiness state than the one currently posted on the INTELBoard blackboard. The order used to prioritize the readiness states is

RED, TEST, YELLOW and GREEN. When an incoming INT report results in the same readiness state as the one currently posted on INTELBoard, the INTELBoard is updated for the sake of recency. The output to the user is still shown each time regardless of whether the INTELBoard is updated. Having completed the discussion of the processing of INT reports it is now time to discuss the functions which actually implement these ideas.

Functions that Process INT Reports

As shown in figure 10 there are only three entities which process the INT message. The figure crowned by an oval shape entitled INTELBoard represents the blackboard and is not a function. METASCHEDULER routes the INT message directly to the INTELHANDLER function. The two remaining blocks named INTELAnalyst and Countries are rule sets as denoted by the encircled RS to the right of their title. INTELHANDLER breaks down the INT message into its constituent parts, and calls on the INTELAnalyst rule set to process the information. INTELAnalyst uses 20 rules to analyze the parts of the INT message and determines the resulting INTELCONDITION.

INTELAnalyst uses the Countries rule set to convert the 1 through 20 country number supplied by the INT message into the actual name of the country. For example, the number 17 corresponds to the country of THEM. Once the INT message is processed, all the SOURCECOUNTRY and REPORTEDCOUNTRY variables are set to the country's name and not the number. INTELAnalyst then displays the output to the user, and determines through its rules whether to update the

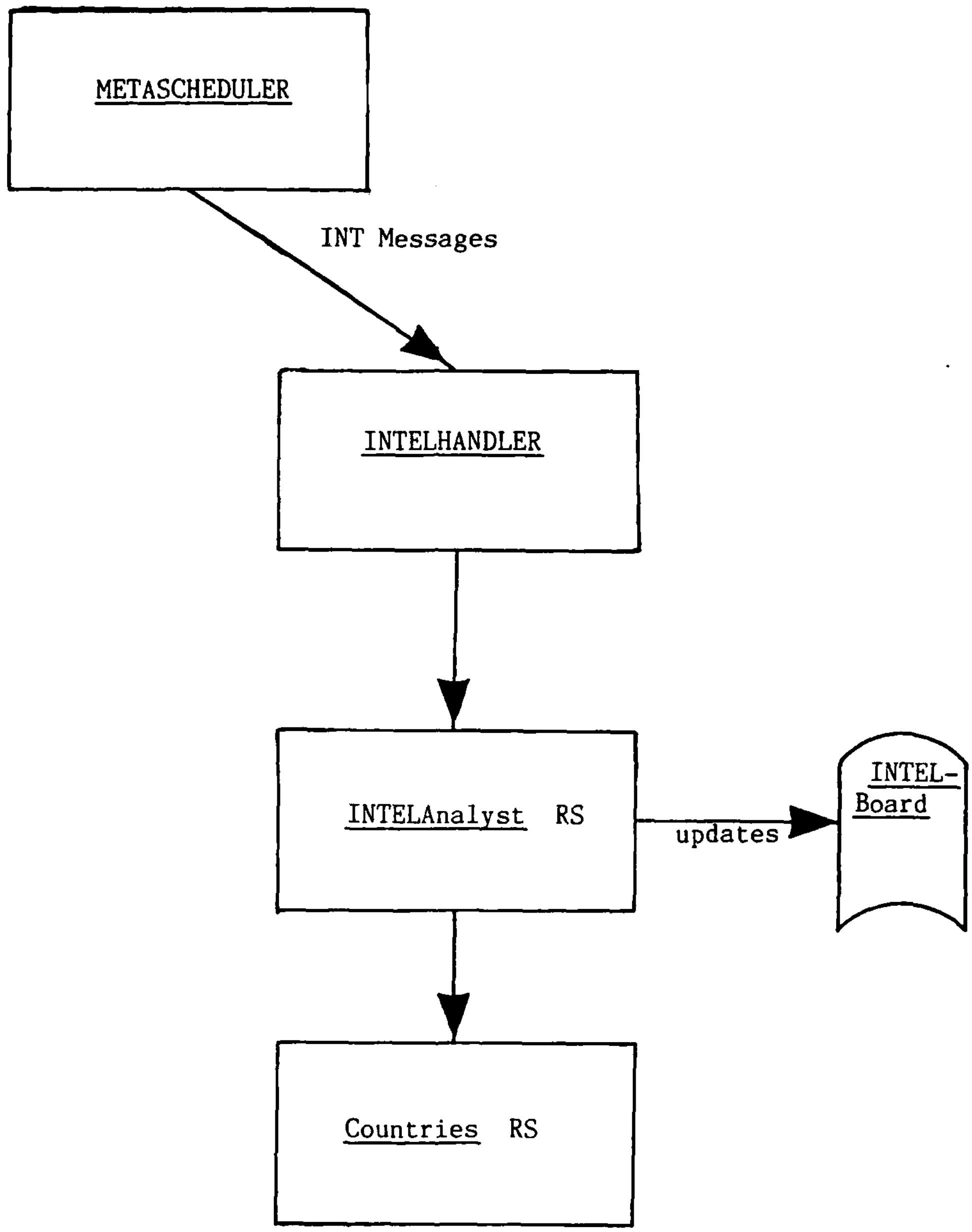


Figure 10. Block Diagram of the Processing of INT Intelligence Messages.

INTELBoard using the priorities described above. If the value of the INTELBoard variables is NIL or of a lower condition INTELAnlayst updates the INTELBoard.

This entire process could have been implemented using one large rule set that would have encompassed all of the functions described above. This was not done here or in the any of the follwing work for several reasons. First of all, the incompatibility of the scenario representation with LOOPS objects indicated that LISP functions would be essential to SENTINEL. Second, it is advisable in a software research effort to keep the functions and rule sets compartimentalized and small to facilitate troubleshooting and debugging. Lastly, many operations with the SIMCOPE messages involved decomposing and constructing list structures and property lists. These operations could have been incorporated in the rule sets as one large THEN statement with multiple LISP commands but this would have obscured the structure and content of the rule set. Therefore, a hybrid structure has been used; LISP functions have been used when extensive list manipulation is required and when it was advantageous to separate these commands from the rule sets, and rule sets were used to describe the heuristics and made use of fairly understandable IF THEN constructs without being burdened with an excessive amount of control knowledge. It must be noted that writing rule sets was a much easier task than developing lengthy LISP functions. SENTINEL could be converted to a much "flatter" non hierarchical structure, if required, by removing the LISP functions or placing them inside the rule sets and constructing larger rule

sets. However, this approach would render further development of SENTINEL more difficult to accomplish and would increase the computation time required. Having covered INT reports, the only other report which is not tied to a launch event, the SYS report, can be discussed.

The SYS System Status Reports

The SYS message is a system status report indicating some type of malfunction at a sensor or a C3 node site. The message indicates if a routine or emergency outage has ocurred, which site is affected, when the outage was reported and the estimated time of return to service. SYS also breaks down the reason for the outage into the following categories: Test, Maintenance, Failure, Sabotage and Unknown. Lastly, the message reports if the current capability of the sensor is FULL, PARTIAL or ZERO. In contrast to the INT reports, there is always a subject actor property list in which to embed all this information. There are five variables that are placed on the property list of the subject actor and they fall into the categories below:

CAPABILITY	[FULL, PARTIAL or ZERO]
REASON	[FAILURE, TEST, MAINTENANCE, SABOTAGE, or OTHER]
RESTORETIME	[reported time at which sensor is restored]
OUTTIME	[time at which the outage occured]
TYPE	[EMERGENCY or ROUTINE]

In a similar fashion to the way INTELBoard was fashioned from INT reports, an SYSBoard blackboard was created which comprised all the variables above. The SYSBoard variables had to be renamed

because instance variables for LOOPS objects cannot use the same names as LISP variables, such as TYPE. There is little information in the OUTTIME variable, so instead SYSBoard uses TYPEOFSENSOR to post whether an ADS satellite, a BSS radar site or a C3 node were being affected. Thus the instance variables of the SYSBoard are the following, with the property list function they represent shown inside the brackets:

OUTAGEREASON	[REASON]
TYPEOFOUTAGE	[TYPE]
TYPEOFSENSOR	[whether it is BSS, ADS or C3]
RESTORATIONTIME	[RESTORETIME]
CAPABILITYOFSENSOR	[CAPABILITY]

The SYSBoard and the property list of the appropriate sensor actor is updated each time a new SYS report is recieved. Updating the SYSBoard on a priority basis was instituted at first but was not pursued because of the C3 network analysis discussed below. In a similar fashion to INT reports, the MWO is merely shown the SYS reports; no acknowledgement is necessary or expected. Thus SENTINEL once again translates the SYS report and the typical output to the MWO is as follows:

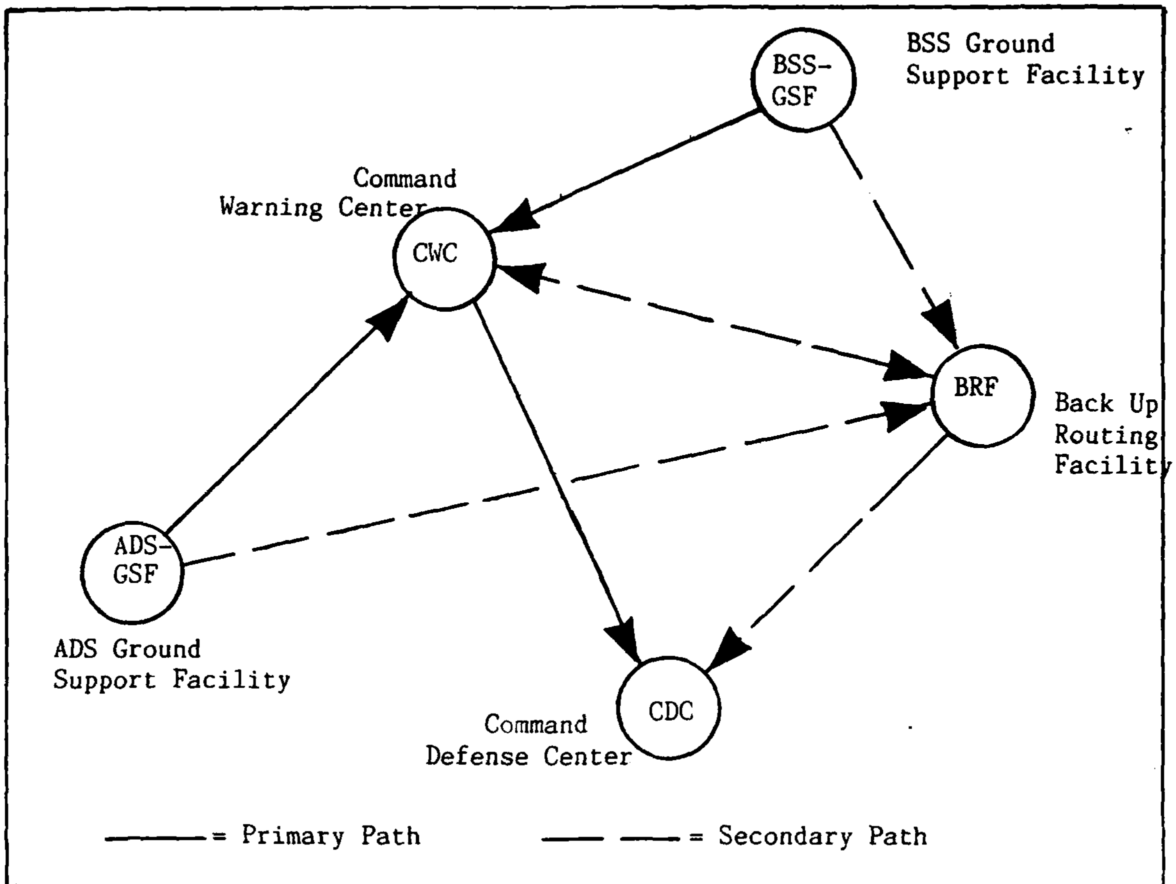
```
SYS Message Acknowledged.
ROUTINE OUTAGE Reported at : BSS2
CAUSE: TEST
CURRENT CAPABILITY OF SENSOR: FULL
TIME OF OUTAGE: 130523
Restoration expected at: 150000
```

Although this method so far has accomplished its primary mission of reporting and recording changes to sensor status, C3 nodes are important enough to warrant further consideration. Since the MWO is

part of the C3 network it is essential that he know what the status and capability of the C3 network is at all times, not just on a site by site approach. Thus, a model for the C3 network used in SIMCOPE needs to be developed.

C3 Network Model and Analysis

Excluding the actual sensors, five sites as illustrated in figure 11 constitute the SIMCOPE C3 network. The Command Defense Center or CDC is the controlling headquarters of this network and is the place to which the MWO sends his reports. The Command Warning Center or CWC is where the MWO is located and where surveillance system data is interpreted and forwarded to the CDC. The ADS Ground Support Facility (ADSGSF) and the BSS Ground Support Facility (BSSGSF) are the control centers for the ADS satellites and BSS radar sites respectively. Finally, the Back Up Routing Facility or BRF permits a second communication path to the entire network. The SIMCOPE manual does not discuss the C3 network any further, but for purposes of analysis of this project, a primary and secondary path of control and communications was assumed for the entire network. The primary path is shown as a solid line in figure 11 and constitutes the path that would normally be followed. The secondary path is shown as a dashed line and shows the alternate lines of communication should the primary ones be impaired. To analyze the C3 network connectivity in the system, four paths were established from the ADSGSF to the CDC and from the BSSGSF to the CDC, as listed in figure 11.



	ADSGSF Paths	BSSGSF Paths
Primary:	(CDC CWC ADSGSF)	(CDC CWC BSSGSF)
Primary:	(CDC CWC BRF ADSGSF)	(CDC CWC BRF BSSGSF)
Secondary:	(CDC BRF ADSGSF)	(CDC BRF BSSGSF)
Secondary:	(CDC BRF CWC ADSGSF)	(CDC BRF CWC BSSGSF)

Figure 11. C3 Network Path Diagram and Listing.

The status of these paths fall into the states of FULL, PARTIAL or ZERO. If the capability of one or more of the sites in a path is less than FULL, then the status of that path cannot be FULL. If one or more sites are PARTIAL and no sites are ZERO than that path is set to PARTIAL. If any site in the path has a capability of ZERO then the status of that path is ZERO. These values are then stored in the SYSBoard as instance variables AFULL (for number of FULL paths to ADSGSF) and similarly APARITAL, AZERO and BFULL (for BSSGSF paths), BPARTIAL and BZERO. Besides keeping track of the status of the C3 network paths, it might also be useful to keep track of the number of sensor sites that are PARTIAL or ZERO status. Out of 7 BSS sites and 2 ADS satellites, the number that are at a PARTIAL capability is recorded in the SYSBoard under AFFSENSOR. Those with a ZERO capability under OUTSENSOR. Care was taken to insure that a sensor that sent out two separate reports was not counted twice. Thus if BSS site 3 first sent out a SYS report listing it with a PARTIAL capability, AFFSENSOR increases by one. If BSS site 3 then sends out a later SYS report degrading its status to ZERO then AFFSENSOR must be decreased by one and OUTSENSOR is increased.

It is also valuable to keep track of the pattern and the cause of the SYS reports received so far and determining what percentage of these messages were EMERGENCY reports. Cause can be established by tabulating how many reports were caused by FAILURE, SABOTAGE and TEST. All these numbers are stored in SYSBoard, which must of course be updated each time a SYS report is received to keep track of the changing total number of reports. To summarize what SYSBoard is

tracking, figure 12 demonstrates what the output of SYSBoard. The actual name of the instance variables used in SYSBoard is shown in brackets to the right. It must be noted that the C3 network analysis is triggered only upon receiving a SYS message that concerns a C3 node, while pattern and cause analysis are performed each time any SYS report is received, through the functions below.

Functions that Process SYS Reports

There are four rule sets and one LISP function that process SYS reports as shown in figure 13. As was the case with INTELHANDLER, SYSHANDLER is a LISP function that decomposes the SYS message it receives from METASCHEDULER. The SYSAnalyst takes the constituent parts in their original numerical form and converts them to such states as FULL or PARTIAL. Thus, an SYS message whose origin is "2" is interpreted to be "BSS site number 2" and an origin of "F" is translated to indicate it concerns the Back Up and Routing Facility (BRF). SYSAnalyst also updates the number of SYS reports and the categories they fall into by posting the values of these instance variables on the SYSBoard. SYSAnalyst updates the property list of the sensor actor each time to embed the five variables: REASON, CAPABILITY, RESTORETIME, OUTTIME and TYPE.

If the SYS report does not concern a C3 node, analysis and integration of the SYS report terminate upon updating the SYSBoard. However, the SYS messages which do concern a C3 node must process through the additional 3 rule sets seen to the left in figure 13. The first rule set, C3Analyst sets each ADSGSF and BSSGSF path as a

```
THE SYSBOARD BLACKBOARD REPORTS THAT:


1 out of 9 sensor sites are OUT          [OUTSENSOR]

with ZERO capability.


2 out of 9 sensor sites are AFFECTED     [AFFSENSOR]

with PARTIAL capability.


4 SYS Reports have been acknowledged.    [OUTTOTAL]

Out of these reports:


4 have been EMERGENCY reports.           [OUTEMERG]

3 have been due to SABOTAGE.             [OUTSABOTAGE]

0 have been due to TESTING.              [OUTTEST]

1 have been due to FAILURE.              [OUTFAILURE]


C3 Network path analysis reveals:


2 ADSGSF paths are FULL.                 [AFULL]
2 ADSGSF paths are PARTIAL.              [APARTIAL]
0 ADSGSF paths are ZERO.                 [AZERO]

4 BSSGSF paths are FULL.                 [BFULL]
0 BSSGSF paths are PARTIAL.              [BPARTIAL]
0 BSSGSF paths are ZERO.                 [BZERO]
```

Figure 12. Output of the SYSBoard. Actual Names of the Instance Variables used in the Blackboard appears in the Brackets on the Right.

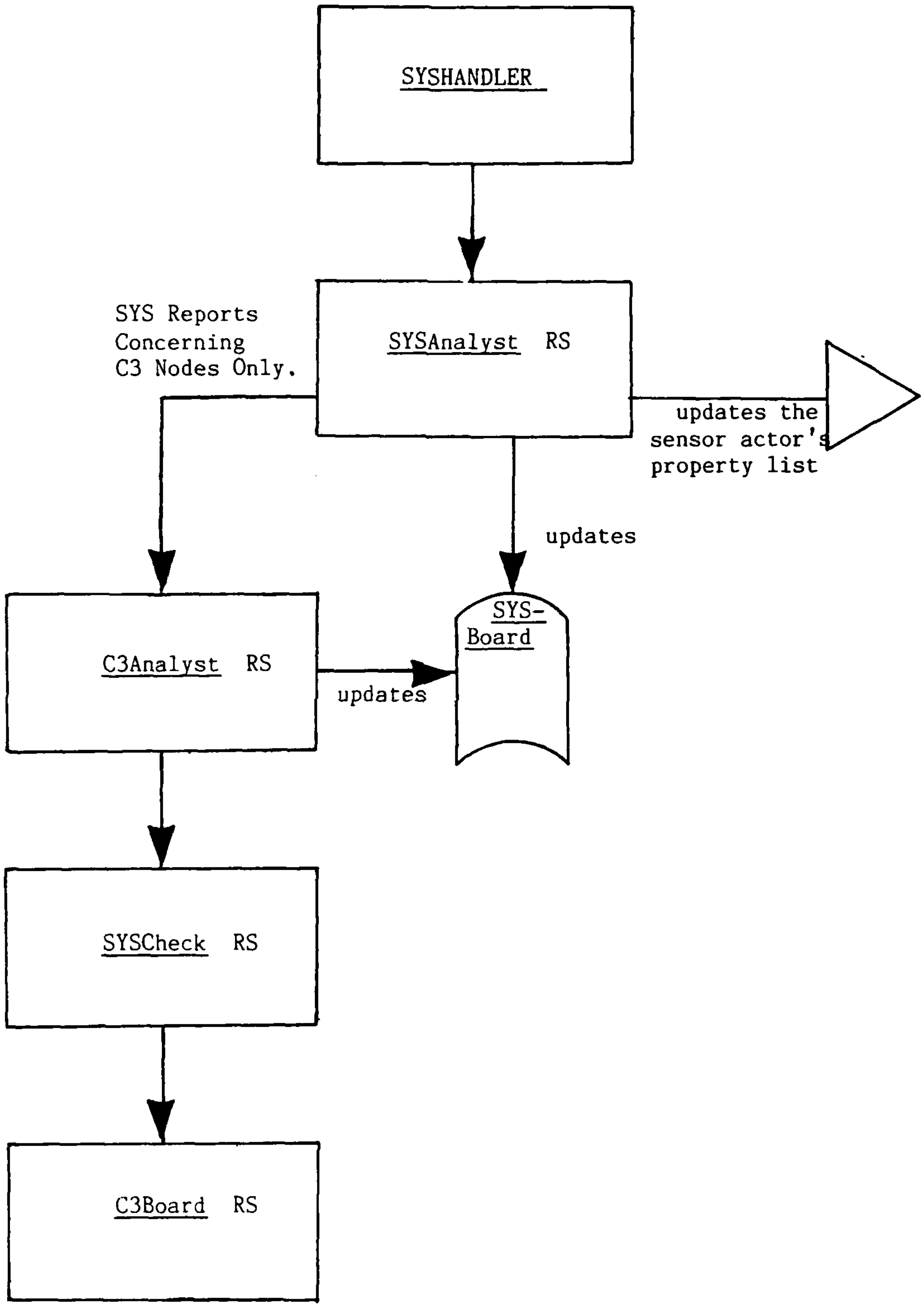


Figure 13. Block Diagram of the Processing of SYS System Status Messages.

list, and composes a "metalist," that is a list of lists, and then calls on SYSCheck with the metalist. SYSCheck uses a While LOOP to look at each separate path on the metalist, and decides whether the status of each path is FULL, PARTIAL or ZERO. Having completed the examining of a path, SYSCheck calls on the C3Board rule set which tracks how many paths fall into each category. Once the path count is updated through C3Board, SYSCheck then deletes the examined path from the metalist and continues to analyze the paths one by one until the list is exhausted. At the conclusion of this process, C3Analyst updates the SYSBoard with the path count maintained by C3Board. Having discussed all the messages that do not concern launch indications, it is now appropriate to discuss the processing of the first report of launch, the ADS-1 message.

Characteristics of the ADS-1 Message

ADS-1 reports were previously discussed as the input to the SCENARIO REPRESENTATION. The most important things they contain are an infrared intensity reading, a latitude and longitude indicating where the launch ocurred and the time when the second pass satellite report is expected to arrive. The MWO must acknowledge ADS-1 messages and fill out a report which asks him various questions such as which region of THEM the launch site is in, was the launch site known, suspected or unknown, and what type of launcher is involved. Finally, the MWO also supplies a subjective assesment of what his confidence level is in believing that this ADS-1 report reflects an actual launch event, rather than spurious data or damaged sensor

readings. The processing of ADS-1 reports is shown on figure 14. Before reaching any further processing, the ADS-1 report goes through the SCENARIO REPRESENTATION. At this point it may be useful to reflect once more on what information is attached to the launch site actor's property list as a result of the polar format processing of the SCENARIO REPRESENTATION:

```
(LAUNCHSITE     [property list, containing]
   LAT          [reported latitude]
   LONG         [reported longitude]
   BSSTAG       [the minimum and maximum headings that can be taken
                from this launchsite and still be detected by a
                particular BSS site]
   HIVALTAG     [range and heading from launchsite to each of the
                high value targets]
   FANCALL      [range and heading to the remainder of the targets])
```

The immediate objective is to assist the MWO to fill out his ADS-1 report by supplying him with the region the missile was launched from and computing a confidence level. Mapping the missile launch site into a region is the task of the Regions rule set which simplifies the task by setting each region into the shape of a rectangle. For example the region of OUTT is any point north of 24 degrees of latitude and east of 17 degrees longitude. The shape of the most complex region, CENTRAL is not described at all. Since all the other regions are described, the launch site is assumed to belong to the CENTRAL region if no other rule is triggered first. Rather than giving the MWO a confidence level on a scale from 0 to 9 as SIMCOPE suggests, SENTINEL computes ADS-1 confidences into five levels: EXTHIGH for extremely high, HIGH, MIXED, LOW and EXTLOW. This task is achieved by the ADS1Confidence rule set which is described next.

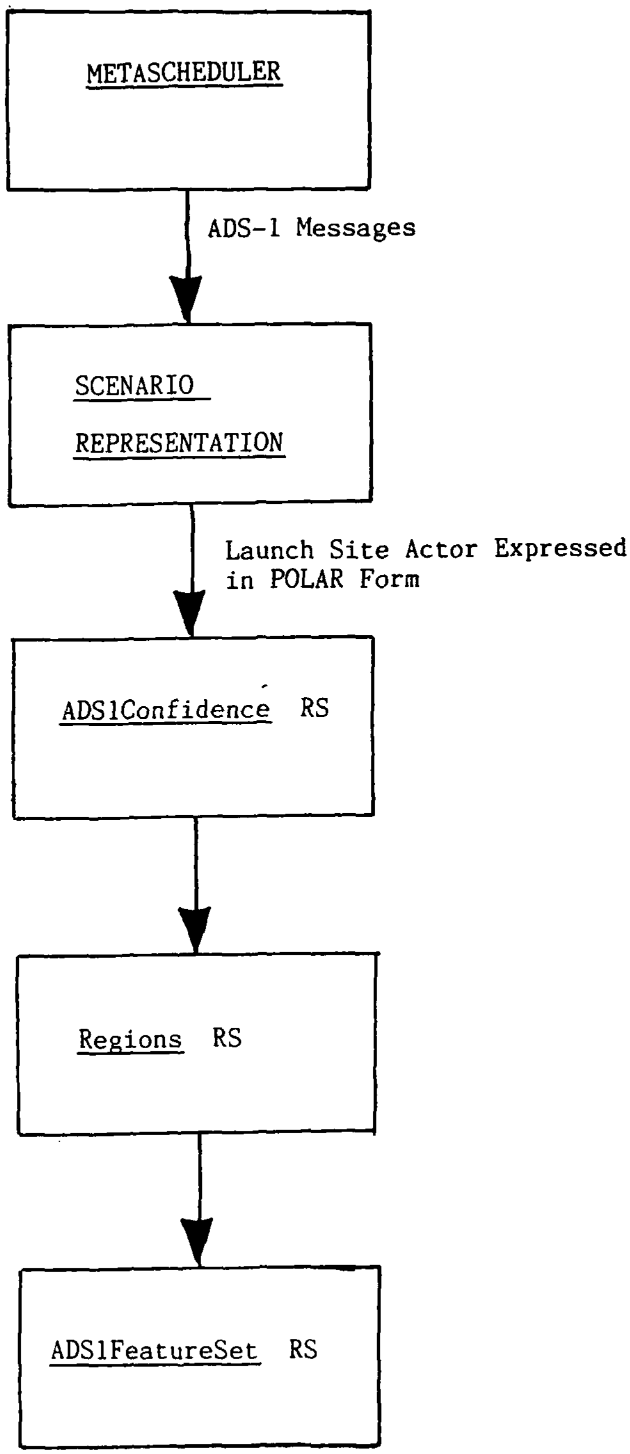


Figure 14. Block Diagram of the Processing of ADS-1 First Launch Report Messages.

Computing the ADS-1 Confidence Level

Compared to the second and third level messages that are discussed later, there is very little information available at the ADS-1 level to decide on the confidence level of the event. It is important to note that receiving an ADS-1 report does not always indicate a missile has been launched. ADS-1 reports can be triggered spontaneously and report raging forest fires or ground explosions as reported launch events (1:68). Infrared readings will be classified into the categories of TYPE1, TYPE2, INDISTINCT or UNKNOWN. INDISTINCT readings are slightly better than UNKNOWN because they occupy a band between TYPE1 and TYPE2 while UNKNOWN readings fall outside the band of readings. The most important factors in determining the confidence level will be the reported infrared intensity, the confidence level of the sensor, and the launch site it has been matched to during the SCENARIO REPRESENTATION processing. If the reported launch event did correspond to a known or suspected missile site, information is available which reports the type of missiles stationed at that site. If the type of missile matches the infrared intensity reading, then there is evidence to confirm a missile launch has taken place. However, if the launch site is unknown then there is no knowledge about the type of missile that could be there. Given purely a heuristic approach to computing the confidence level, one is tempted to write many rules such as:

```
IF   Sensor Confidence is HIGH     AND
     Infrared Reading is INDISTINCT
THEN ADS1Confidence level is HIGH
```

There are relatively few difficulties with this approach unless many factors are combined together; in those cases writing a rule to cover every single occurence becomes a repetitive and monotonous process. Using the above rule as a starting point, it becomes difficult in this form to easily combine the conclusion of the above rule with other factors such as the INTELCONDTION and the reported capability of the satellite which originated the report. Instead of progressively combining more information using such abstraction levels as HIGH or LOW, numbers were used. Numeric approaches if chosen carefully will render the same results as the more cumbersome combining of abstract states. To exhaustively cover every single combination of states in a "brute force" approach would require over 400 rules, whereas using numerical processing methods requires only 30 rules. It is much easier to map an INDISTINCT infrared level to a "1" and a HIGH Sensor Confidence level to a "2," add the two together and map the sum "3" to an overall confidence level of HIGH. Besides the ease of manipulating numbers, the theory of heuristic reasoning by using endorsements developed by Paul R. Cohen supports the ranking of evidence in different manners and specifically includes the possibility that such endorsement levels can be numerical measures (5:53). Using the terminology developed by Cohen, one could say that the evidence for supporting the hypothesis that a launch event has occurred is higher for a HIGH sensor confidence level which then results in a higher degree of belief of evidence in this factor which is a "3." Conversely one can state that the INDISTINCT infrared level does not support the hypothesis nearly as

well and thus receives its "2" ranking. One other important benefit from this approach is the ease with which the output of the system can be changed. By merely changing the ranking of evidence or tinkering with how the sum is interpreted at the end, the confidence level can be varied greatly just by changing a handful of numerical assignments.

The task of ADS1Confidence is to take five separate factors and map them to a number depending on how well they support the hypothesis of an actual missile launch. These five numbers are added up together and the final sum is interpreted over a range of values to result in the confidence levels of EXTHIGH, HIGH, MIXED, LOW and EXTLOW. The five factors ADS1Confidence uses are: the infrared intensity and the reported capability of the satellite making the report, the intelligence condition, status of the launch site (is it known or suspected?) and the sensor capability as reported in its property list. At this point, the MWO can fill out the ADS-2 report with the assistance of SENTINEL.

```
ADS1 INDICATION OF LAUNCH RECEIVED:

    ADS PASS 1   EVENT1

1. DETECTED: 143500          [time of ADS detection]
2. ADS:  ADSSOUTH            [which satellite made the report]
3. REGION: CENTRAL           [launch site in this region of THEM]
4. LAUNCHER TYPE: TYPE2      [infrared intensity reading results]
5. CONFIDENCE: HIGH          [resulting ADS-1 confidence level]
```

It now becomes crucial to incorporate an explanation capability, a central feature of any expert system into SENTINEL.

Explanation at the ADS-1 Level

The ADS-1 explanation capability is rudimentary simply because at this level there is not that much information available. Only the five factors which were used in computing the ADS-1 confidence level directly affect the reliability of the report. Therefore, the ADS-1 explanation capability consists solely of a stating of the five factors that led to the confidence level. The MWO is asked to respond yes or no (actually T or NIL) to whether he wishes to see more information. Requiring the MWO to respond is appropriate since he must respond to ADS-1 messages. What the MWO sees following the output shown above is as follows:

```
Further explanation can be obtained by typing a T,
otherwise type NIL.
T
SENSOR Confidence is: HIGH
SENSOR Capability is: FULL
IR Readings classify event as: TYPE2
Current INTEL condition is: YELLOW
As reported by INTEL sources.
```

For the second and third level messages, this type of simplistic explanation will not suffice. What is required is a method of displaying all the relevant data if required but not overwhelming the operator by simply displaying a mass of numbers and statistics each time he requests an explanation. It would be more useful to abstract that mass of data to a higher level such as "INTEL conditions suggest ATTACK", however that is one of the main concerns of the next chapter. Before turning to chapter 4, the last ADS-1 rule set entitled ADS1FeatureSet must be discussed.

ADS1FeatureSet

In describing and updating launch reports, it is necessary to have some sort of structure to track a missile launch as it proceeds through the various phases of sensor detection such as ADS-1, ADS-2 and BSS stages. The first impulse is to embed the ADS-1 information onto the launch site actor's property list. The problem with this approach is that the same launch site could account for multiple launches and thus ADS-1 reports. To avoid destroying this information, a structure independent of the launch site must be used to embed this information. This structure is also a property list which is entitled an EVENT. An event counter is maintained to insure each event property list created is unique. Thus the first ADS-1 report results in EVENT1, the second one EVENT2 and so on. METASCHEDULER actually creates the EVENT by using the event counter, however the EVENT is not completed until ADS1FeatureSet is processed.

One could take all the information provided by the ADS-1 report and blindly embed it on the EVENTLIST using such variables as SENSORCONFIDENCE, INFRAREDINTENSITY and so on. However, this approach tends to obscure the essential data amid a lot of less valuable information. A far more profitable approach is to use the concept of a "feature set" to isolate the essential characteristics of the launch report into the fewest yet most distinct set of characteristics. Feature sets have been used widely in pattern recognition tasks to identify unknown objects by computing the distance between the unknown object and the characteristics of known

models (22:369). In the case of SENTINEL, the concept of a feature set will be used to compute the difference or distance between the unknown launch event and the known characteristics of such hypotheses as TEST, HOSTILE and UNKNOWN. At the completion of processing for the ADS-1 report, the end result is the following event list and feature set, where the leading P1 indicates the first pass, and the items in the brackets are a short explanation of what each variable contains.

```
(EVENT1  [name of the property list]
  P1TIME [time at which the ADS-2 report for this event is expected]
  P1LS   [the launch site actor which originated this report]
  P1CONFIDENCE  [the level of confidence]
  P1IRTYPE      [result of the infrared reading]
  P1SENSOR   [combined output of sensor confidence and capability] )
```

The first two items of the property list are not really part of the feature set but are essential to maintain. P1TIME is used to match the followup ADS-2 report to the same missile launch report, and thus the same EVENT. P1LS is the launch site actor this ADS-1 report and EVENT have been matched to and is the key to all the SCENARIO REPRESENTATION processing. The remaining three items constitute the feature set of the ADS-1 report.

P1CONFIDENCE is the HIGH or MIXED confidence rating determined by ADS1Confidence and represents a useful abstraction of the end result of all the information for this launch event. P1IRTYPE is carried over directly from the ADS-1 report, because it cannot be easily combined with other factors without losing information in the process. The final item, P1SENSOR represents the combination of sensor site confidence on the ADS-1 report with the sensor

capability of FULL, PARTIAL or ZERO. The resulting P1SENSOR is also a 1 through 9 number which results from the following process. First the sensor confidence level is abstracted to the following ranges: 9 through 7 results in HIGH, 4 through 6 is MEDUIM and anything below 6 is LOW. P1SENSOR is then set through several IF THEN rules. For example:

```
  IF  SENSOR CAPABILITY  is:  FULL     AND
      SENSOR CONFIDENCE RANGE is: MEDUIM
THEN  P1SENSOR is: 8
```

Using numbers makes it easy to compare different sensor configurations, as a P1SENSOR rating of 8 is better than a sensor rating of 7, since the rating of 7 occurs when sensor capability is PARTIAL and confidence is HIGH. At the same time this approach takes advantage of the discrete nature of the SIMCOPE variables to produce a unique output which can always be broken down into a sensor capability and a sensor confidence. This permits a higher level of abstraction to be attained which is easier to compare and propagate while still being able to use its constiutents if needed. Producing this feature set, which represents as much information as possible by using the fewest number of characteristics, brings the processing of ADS-1 messages to completion. ADS-2 second level messages are processed in the next chapter.

IV. Processing of the Second Pass Satellite Report

The ADS-2 message represents the second, confirmatory indication of a missile launch from the ADS satellite in the SIMCOPE scenario. When the ADS-2 message is coupled with the appropriate ADS-1 message, a large amount of information becomes available. By using a blackboard architecture, SENTINEL infers and reasons from this information in a way which is consistent with other expert systems. SENTINEL must also determine whether the launch event in question represents a TEST, HOSTILE, UNKNOWN or UNRESOLVABLE event. This is determined using the statistical concept of a feature set, and measuring the distance to each hypothesis to the EVENT in question. Before these other tasks can be accomplished, the ADS-2 report must be matched up with the appropriate EVENT.

Matching ADS-1 and ADS-2 Messages

The information presented in the ADS-1 report and captured within the property list of the EVENT includes: the launch latitude and longitude, the infrared intensity reading, the confidence of the sensor site and when the ADS-2 report can be expected. By contrast the ADS-2 message yields a heading, a missile type, sensor site confidence, what BSS site will detect the missile and a predicted time for BSS site detection. ADS-2, like all the other SIMCOPE messages, give a PRESENTIME which is the time at which the received message was generated. The predicted time of appearance for the ADS-2 report• which is the P1TIME of the EVENT can be used with

PRESENTIME to match the ADS-2 report. For example, if EVENT3 has a P1TIME of 2130 hours and an ADS-2 report is received at 2131 hours, strong circumstantial evidence links this report to EVENT3. Note that the number of ADS-2 messages will be less than the number of EVENTs since the ADS-1 messages which create the EVENTs have a much higher possibility of being a false alarm (1:47). SENTINEL cannot process an ADS-2 report which cannot be matched to an EVENT, since no longitude and latitude coordinates would be available and therefore no available polar representation of the SCENARIO. If such an event were to occur, SENTINEL would avoid performance degradation difficulties by announcing to the user that :

NO ADS1 EVENT MATCHES THIS ADS2 REPORT
SPURIOUS ADS2 REPORT SUSPECTED
SENTINEL DECISION AID CANNOT PROCESS THIS REPORT

By using this technique the MWO can then be made aware of what SENTINEL cannot aid him with and can deal with it as he chooses.

To begin the matching process, SENTINEL first checks the compatibility of satellite coverage with the P1LS or launch site identification of the EVENT. If EVENT3 has a P1LS whose latitude is at 35 degrees, and the ADS-2 report in question originated from the southern ADS satellite, they will not be matched since the southern satellite's region of coverage does not normally encompass 35 degrees. Once compatibility of satellite coverage is ascertained, SENTINEL matches the ADS-2 report to the EVENT whose P1TIME is closest to PRESENTIME. SENTINEL will not automatically process EVENTs whose ADS-2 report is more than 10 minutes overdue. Since the

longest range SIMCOPE missile has a maximum flight time of 10 minutes and a report is more than 10 minutes overdue one can assume that the missile is no longer still in the air. When a match takes place, the EVENT in question becomes the EVENTMATCH variable and P1TIME, which indicated the predicted time of reception for the ADS-2 report, is removed from the EVENTMATCH property list and P2TIME, which indicates when the BSS radar report is expected, is added to the property list. By examining the property list of an EVENT one can always tell what stage of processing it has been through, for if EVENT3 has been matched to an ADS-2 report, it will have NIL for a P1TIME value but have a number for the P2TIME slot. Having coupled the ADS-1 and ADS-2 information, one has available a launch site location, a missile heading as well as a second infrared intensity and sensor confidence rating. Determing a feature set and confidence level for ADS-2 processing is the next objective.

Initial Processing of ADS-2 Message

As shown in figure 15, the initial processing of ADS-2 messages is quite similar to the ADS-1 process. METASCHEDULER routes the ADS-2 message directly to ADS2HANDLER, which is a LISP function that decomposes the ADS-2 report into its constituent parts, and then attempts to match an eligible EVENT from the EVENTLIST with the ADS-2 report. If a match is made, ADS2Confidence and the ADS2FeatureSet rule set are invoked to determine a second pass confidence or P2CONFIDENCE and a feature set which can be added to the EVENTMATCH. If no match is found, ADS2HANDLER calls on EVENTCHECK, which shows

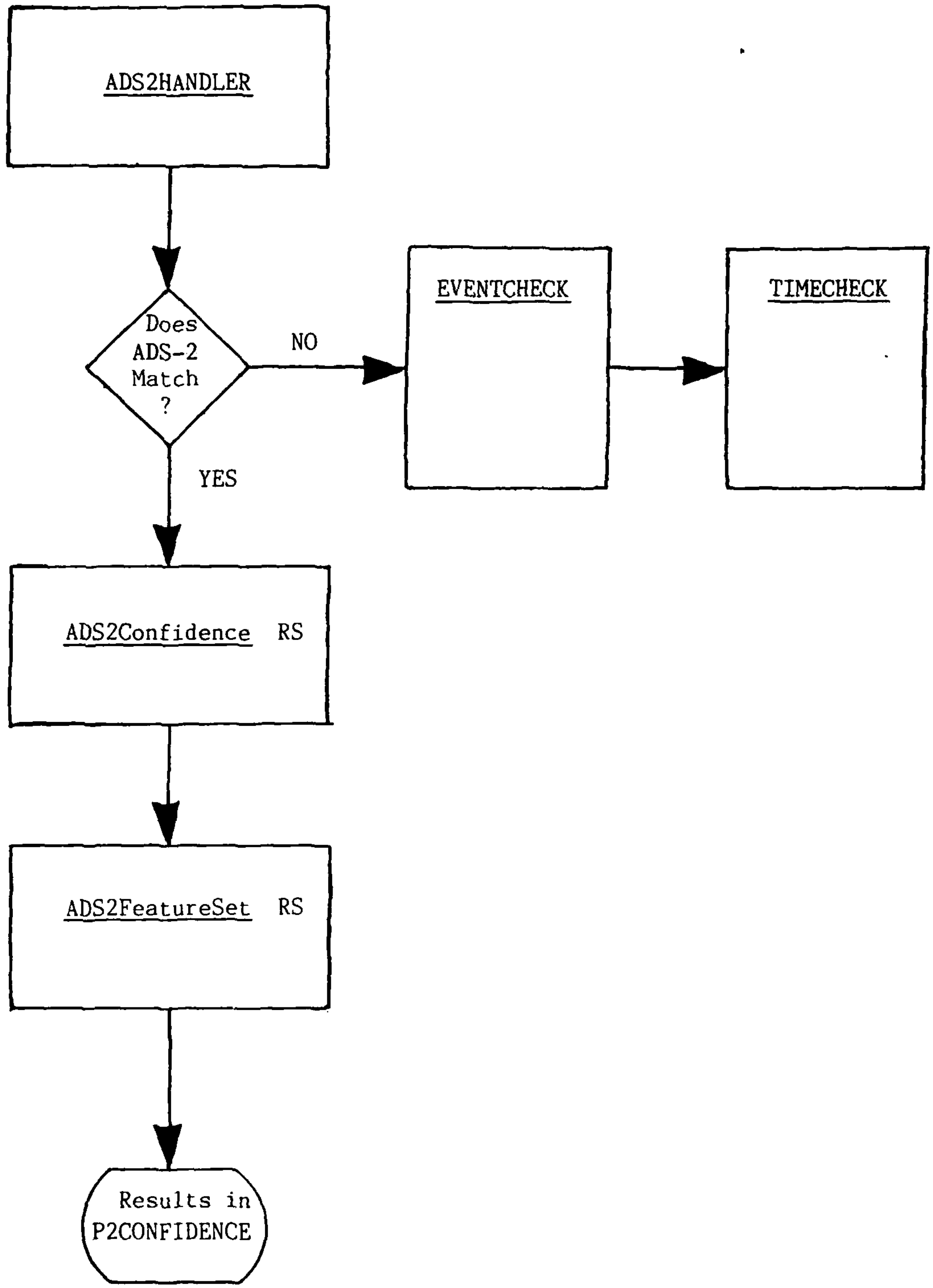


Figure 15. Initial ADS-2 Processing.

three lists to the MWO; first the EVENTLIST of all EVENTs established so far, than a list of all the EVENTs which have not yet been matched to an ADS-2 report, as well as the ADS-2 reports whose satellite coverage is incompatible. EVENTCHECK also calls on TIMECHECK to locate any EVENTs whose ADS-2 reports are overdue by 10 or more minutes, and then asks the user if he wishes to match any of the previously unmatched reports, or delete the overdue messages as reported by TIMECHECK. During the course of normal SIMCOPE operations, it is unlikely that the TIMECHECK and EVENTCHECK functions will be used, however they are quite effective for training and troubleshooting.

Having matched the ADS-2 message to an EVENT, the ADS2Confidence rule set is invoked to compute an ADS-2 level of confidence. ADS2Confidence uses the P1CONFIDENCE slot of the EVENT property list as a starting point, which is reduced or increased by examining the infrared intensity reading and the sensor confidence level expressed in the ADS-2 message. Numerical methods are used to cut down on the number of rules required. P1CONFIDENCE is first interpreted as a SUM in a process that is the reverse of ADS1Confidence. The higher this number, the more certain the confidence level since EXTHIGH results in a 5, a confidence of MIXED is a 3 and so on. At this point ADS2Confidence calls on ADS2FeatureSet to reduce this information to a feature set, which contains only the four elements listed below.

P2TIME	[time at which BSS radar report expected]
P2CONFIDENCE	[confidence level resulting from ADS2Confidence]
P2TYPE	[relates ADS-1 infrared to ADS-2 infrared]
P2SENSOR	[similar to P1SENSOR, 1 through 9 number]

Most of these items are quite similar to the ADS-1 feature set, such as the time expressed in P2TIME and the five level confidence (EXTHIGH, HIGH , MIXED, LOW, EXTLOW) expressed previously, as well as P2SENSOR which combines the second pass sensor confidence and sensor capability into a single numerical rating as was done for P1SENSOR. However, P2TYPE is different and represents a more abstract level of information, taking on 3 values: MATCH, NOMATCH, MISMATCH. MATCH indicates that the ADS-2 infrared intensity corresponds exactly to the P1IRTYPE. If EVENT1 has a P1IRTYPE value of TYPE1 and ADS-2 also indicates TYPEI then P2TYPE is MATCH. On the other hand, if the ADS-2 infrared intensity differs from P1IRTYPE but for the better, NOMATCH is the result. If P1IRTYPE is INDISTINCT but the ADS-2 reading corresponds to TYPE2, this results in NOMATCH. Finally, MISMATCH indicates the opposite case; the one of a certain rating such as TYPE1 or TYPE2 degrading into an INDISTINCT or UNKNOWN reading. When ADS2FeatureSet has determined P2TYPE and P2SENSOR, ADS2Confidence can now determine the P2CONFIDENCE. Using P1CONFIDENCE as an input, ADS2Confidence examines the results of ADS2FeatureSet to indicate whether the evidence further supports or denies the confidence level expressed in ADS-1. A higher sensor rating, or a P2TYPE of MATCH or NOMATCH result in higher confidence levels, while lower sensor ratings and MISMATCH indications tend to lower the P2CONFIDENCE. It may be useful to reflect at this point what the entire event list contains following the ADS-2 match and what values each parameter can take:

```
(EVENTNUMBER
      P1TIME          [NIL, was deleted to indicate match]
      P1CONFIDENCE    [EXTHIGH HIGH MIXED LOW EXTLOW]
      P1SENSOR        [1 thorugh 9 numerical rating]
      P1IRTYPE        [TYPE1  TYPE2  INDISTINCT  UNKNOWN]
      P2TIME          [to be used for BSS match]
      P2CONFIDENCE    [EXTHIGH HIGH MIXED LOW EXTLOW]
      P2SENSOR        [1 through 9 numberical ADS-2 rating]
      P2TYPE          [MATCH  NOMATCH  MISMATCH] )
```

In order to reason with the attributes of the EVENT, it is necessary to have a way of storing, analyzing and displaying this data. Adding more information to the EVENT would detract from the goal of keeping the feature set as small as possible, so another knowledge representation strategy must be used.

Blackboard Model Proceesing

The concept of a blackboard model was pioneered by the HEARSAY expert system which featured a global data structure known as a blackboard that was accessed by all the knowledge sources. Blackboard model systems have been used for the interpretation of noisy signals (HEARSAY: Erman and Lesser, 1975), and data from sensors (CRYSALIS: Englemore and Nii, 1977) and an opportunistic problem solver (OPS: Hayes-Roth, 1978) (4:III-336). This type of architecture can be used for representing information at different levels as it permits specialist subprograms to post their hypotheses on a blackboard where it is available to all the other specialists as well. HEARSAY presented the idea of indpendent knowledge sources cooperatively solving a problem by posting hypotheses on a global blackboard data structure. Other aspects of SENTINEL which can be found in a blackboard model is the event driven control structure

presented in CRYSALIS which uses an event list to keep track of hypotheses as they proceed through different planes or layers of reasoning in the protein crystallography domain (4:II-126). SENTINEL owes a distinct debt to blackboard architecture since it uses an event driven control structure and an event list, and will use blackboards to generate a reasoning capability and a two layer explanation capability.

What is proposed for SENTINEL is a group of specialist subprograms that each post a hypothesis about whether the data they have analyzed supports the contention that an actual missile launch has ocurred and not a system or testing error. Rather than using a central blackboard, five separate and distinct blackboards will be maintained to make it easier to recall what data is posted on each blackboard. This community of blackboards includes the previously discussed INTELBoard and SYSBoard as well as three others. MATCHBoard indicates the credibility of the sensor at the time of the report, the TRENDBoard indicates whether the trend of evidence is going up or down in going from ADS-1 to ADS-2, and DETECTBoard indicates the likelihood of BSS sites being able to track the missile event in question. Each of these five blackboards will contribute to the reasoning and explanation process by having an EXPNUMBER, which is a state on a scale of 1 to 4 indicating how strong belief is in an actual launch event, and a COMMENT for the benefit of the MWO which represents an abstracted description of the state listed in EXPNUMBER. Therefore, subprograms will post their determinations on each blackboard so that SENTINEL can reason from

their postulations. The COMMENT section of all five blackboards will serve as a foundation for the SENTINEL explanation capability. For example, if INTELBoard analysis reveals a GREEN condition with no alarming incidents posted, the value of EXPNUMBER would be 1 since that does not strongly support the hypothesis of an actual launch event taking place and an appropriate COMMENT would be: "INTEL reports no indications of an ATTACK." On the other hand, an INTELBoard with a RED condition would have an EXPNUMBER of 4 and the COMMENT would be: "INTEL reports STRONG POSSIBILITY of ATTACK." Note that the abstraction listed in the COMMENT and the EXPNUMBER is less precise than a recital of all the past INT reports for the last half hour would be. The key to understanding the explanation capability of SENTINEL is to understand the four levels of information available in SIMCOPE as shown in figure 16.

The first level is the "raw" SIMCOPE message data which must be deciphered and interpreted. Once a message such as an SYS message recording an EMERGNECY outage at BSS site number 2 due to a TEST is received, such information is also conveyed in the property list of the appropriate actor. In this case the BSS#2 property list is updated to show an outage has taken place that results in a capability of FULL until 1500 hours. The second level consists in capturing the information in the property list of the appropriate actor. The third level is the tabulation of this information in the blackboards. For example the SYSBoard would keep track of how many SYS outage reports have been received, and out of those which ones were EMERGENCY ones, while the DETECTBoard would record the BSS

```
                     SIMCOPE MESSAGES

FIRST LEVEL      ((CWC (SYS (R 2 S T) 130523) 1500) F)
                 ((CWC (ADS ((S1 142718) 8) -29) 13) (142915) 8)


                     ACTOR's PROPERTY LISTS

SECOND LEVEL    (BSS#2  CAPABILITY [FULL]   REASON [TEST]
                     OUTAGETIME [130523]   RESTORATIONTIME  [1500] )

                (LS6   TYPE [RED1]  LAT [29.5]  LONG [13.5]
                     BSSTAG    FANCALL    HIVALTAG  [VARIOUS] )


                    BLACKBOARD PROCESSING

THIRD LEVEL     (SYSBoard  15 SYS Messages Recieved
                           10 EMERGENCY Reports
                      2 SENSORS are OUT with ZERO CAPABILITY
                      1 SENSOR IS AFFECTED with PARTIAL CAPABILITY)

                (DETECTBoard
                        BSS#2  BSS#3    WILL DETECT.
                        BSS#4           COULD DETECT.
                        BSS#5           WILL PROBABLY NOT DETECT.)


                    ABSTRACTION LEVEL

FOURTH LEVEL     "MANY SYS REPORTS RECEIVED         EXPNUMBER = 2
                  OVERALL SYSTEM STATUS:  POOR"

                 "POSSIBILITY OF BSS DETECTION:     EXPNUMBER = 4
                  VERY STRONG"
```

Figure 16. The Four Levels of Information Representation
Used in SENTINEL.

sites which would be in a position to detect the missile heading. The fourth and highest level of abstraction is the one represented by the COMMENT and EXPNUMBER; a less precise yet more useful way of capturing information at a glance. Compare the following:

```
20 SYS REPORTS RECIEVED, OUT OF THESE REPORTS
      10 were EMERGENCY reports
       2 were due to SABOTAGE
       3 were due to FAILURE
       3 sensor sites are AFFECTED with PARTIAL capability.
```

to the less precise but shorter:

```
MANY SYS REPORTS received: OVERALL SYSTEM STATUS: IMPAIRED.
```

Certainly the first form is more accurate and specific and does not have any "noise" in it, however it is harder to understand quickly and could easily drown the MWO in another wave of information; precisely the problem SENTINEL is trying to alleviate. The second form is less precise and has more "noise," yet it succinctly represents the general state of affairs for SYS messages. The comparison of these two methods leads to the two layer explanation capability of SENTINEL. The top layer represents the COMMENTs which are meant to be short "reminders" of a particular situation. In most cases, the MWO will need or only have the time for this level of explanation. If the MWO is not satisfied with this layer of explanation, the second layer which is the third level of information representation, or the tabulation of the blackboards can be shown.

INTELBoard and SYSBoard

These first two blackboards were introduced to the reader in chapter 3 while discussing the processing of INT and SYS messages. The only modification necessary to the INTELBoard and SYSBoard were the addition of a rule set INTELComment and SYSComment respectively, which determines the COMMENT and EXPNUMBER of these blackboards. If SENTINEL has just been initialized, no INT or SYS reports may be available. When no INT reports have been received the EXPNUMBER is 0 and the COMMENT is NIL since no information is available. In the absence of information to the contrary, no SYS messages indicate a perfectly well functioning sensor system with an EXPNUMBER of 4, since if the sensor and C3 network are completely unimpaired there is more confidence that the launch event they are reporting is indeed ocurring and is not the result of damaged or faulty sensors. Having discussed the two blackboards that are independent of ADS-2, it is now appropriate to describe the other three blackboards which are directly tied to the ADS-2 report.

MATCHBoard, DETECTBoard and TRENDBoard

As shown in figure 17, the three remaining boards are filled by five rule sets. The first rule set invoked by ADS2Confidence is SensorCheck. SensorCheck reports on the credibility of the sensor making the ADS-2 report. Any reported outages recorded on the property list of the sensor actor are examined and are posted on the blackboard. In addition to the sensor information, MATCHBoard also stores additional information about the launch event such as if the

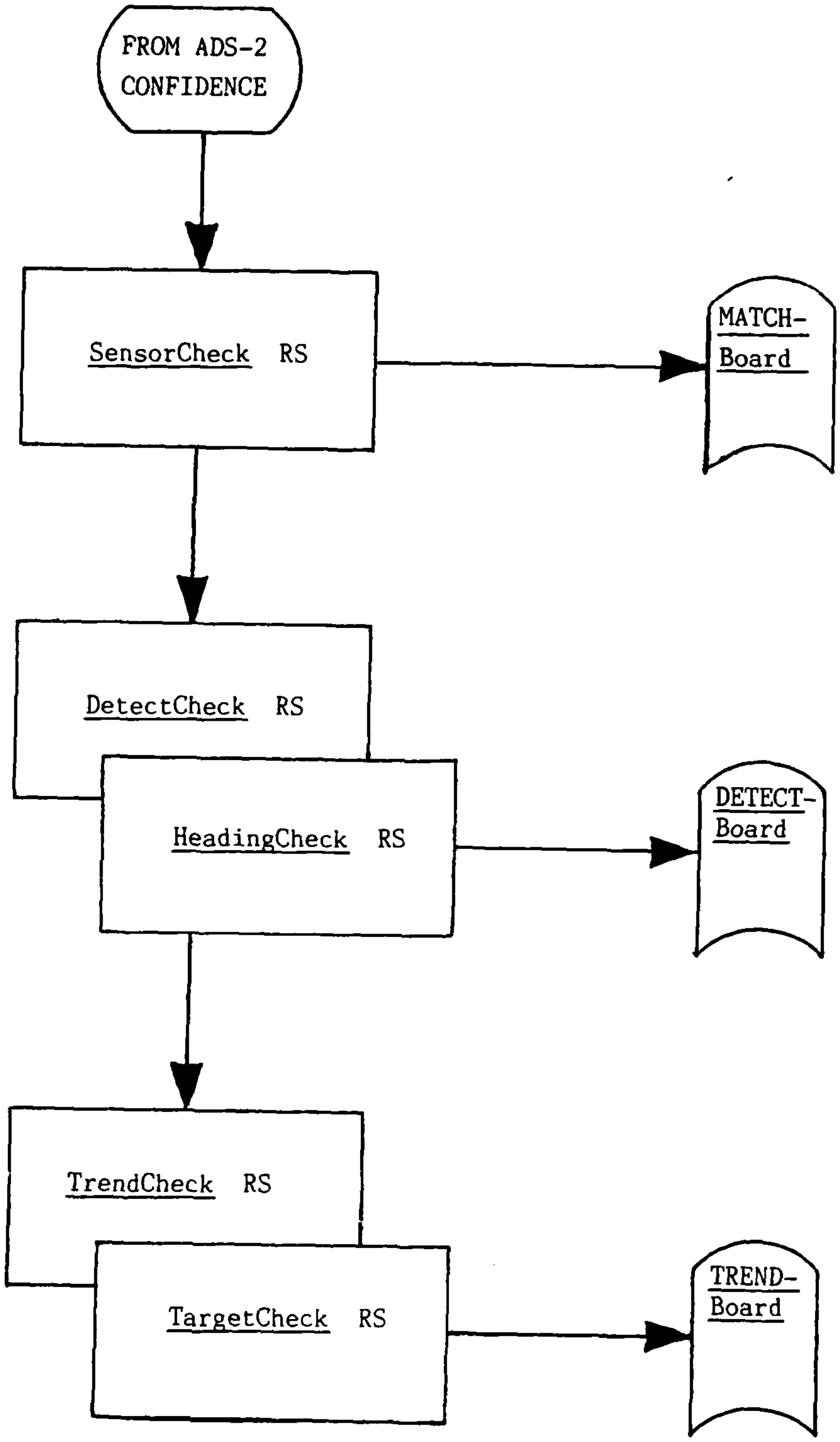


Figure 17. Blackboard Processing Following Initial ADS-2 Processing.

launch site is KNOWN, UNKNOWN or UNSUSPECTED. All of the instance variables maintained in MATCHBoard are shown below along with a short explanation of what they represent:

RPTSENSORCAP	[capability of the sensor: FULL PARTIAL or ZERO]
OUTAGEREASON	[outage reason: SABOTAGE, FAILURE, TEST or UNKNOWN]
LAUNCHSITE	[its status: KNOWN, UNKNOWN or SUSPECTED]
LSHEADING	[records the reported heading of the missile]
LSBSS	[ADS-2 prediction of what BSS sites will detect]
P2LS	[the launch site it has been matched to, such as LS7]
LAUNCHERTYPE	[type of missile: TYPE1 or TYPE2]
EXPNUMBER	[abstracted general state of the MATCHBoard]
COMMENT	[for first layer of explanation, tied to EXPNUMBER]

Note that the LAUNCHERTYPE is assumed to be TYPE2, the longer range missile with a maximum flight time of 10 minutes and maximum range of 2,650 kilometers, unless proved otherwise. LAUNCHERTYPE can be TYPE1 only if the infrared reading of both the ADS-1 and ADS-2 messages determine it to be a TYPE1 missile, which has a maximum flight time of 5 minutes and thus a range of 1,325 kilometers. Having completely filled the MATCHBoard, DetectCheck rule set is invoked next. DetectCheck concerns itself with the detection of the missile event by a BSS site. Using the missile HEADING given in the ADS-2, DetectCheck calls on HeadingCheck to process the BSSTAG of the launch site actor. One may recall that the BSSTAG contains the maximum and minimum headings that a missile can take from that given site and still be detected by the BSS site. Thus BSSTAG takes the following form:

```
(LS7   [launchsite actor property list]     BSSTAG
    ((LS7 MAXPT MINPT) (BSS1 123  142) (BSS2 136 150) and so on.))
```

In the example given above, a missile launched from LS7 will be

detected by BSS site number 1 if the heading lies between 123 and 142 degrees. Even though HEADING is given as a single number, TYPE2 missiles can manuver up to 5 degrees from the heading reported in ADS-2. If a TYPE2 missile heading of 125 degrees is given for example, that missile could actually strike any target between 120 and 130 degrees. Thus, it is not sufficient to merely state if the heading will be detected by the BSS site in question if the heading lies between the reported MAXPT and MINPT. Instead, various lists must be maintained: the WILLDETECT list of those BSS sites, where the HEADING does lie between MAXPT and MINPT, the CLOSEDETECT list of those sites whose range of coverage is up to 5 degrees away from detecting the HEADING and the FARDETECT list of those sites whose coverage lies up to 10 degrees away from the HEADING. In some cases, the HEADING will deviate more than 10 degrees from any BSS site of coverage, and SENTINEL would state:

NO BSS DETECTION POSSIBLE, CLOSEST approach will be to
BSS SITE 7 with a deviation of 23 degrees.

The entire COMMENT associated with DETECTBoard is shown above, where NEARPT is BSS site 7 and the NEARDIFF represents the closest deviation from a BSS zone of coverage that lies further than 10 degrees. The EXPNUMBER associated with this COMMENT would be a 1 since it does not support the contention that a hostile missile launch has ocurred since it is proceeding along a heading that does not take it over US. A DETECPT is also needed which is used to complete the ADS-2 report. When responding to the ADS-2 report the MWO is asked only for a single BSS site, not a list such as

WILLDETECT. Therefore, DETECTPT is the BSS site whose middle zone of coverage (the average between MAXPT and MINPT) lies most closely to missile heading. DetectCheck and HeadingCheck have completed their task when the DETECTBoard variables below are filled out.

WILLDETECT	[those BSS sites where heading is in zone of coverage]
CLOSEDETECT	[BSS site coverage up to 5 degrees away from heading]
FARDETECT	[BSS site coverage up to 10 degrees away from heading]
NEARPT	[BSS coverage closest to heading, but further than 10 degrees]
NEARDIFF	[the difference between heading and coverage of NEARPT]
DETECTPT	[BSS site number needed for ADS-2 report]
STATUSDETECTPT	[DETECTPT capability: FULL PARTIAL or ZERO]
EXPNUMBER	[abstracted state describing BSS detection opportunities]
COMMENT	[used in first layer of explanation, tied to EXPNUMBER]

Only TRENDBoard which is filled by TargetCheck and TrendCheck remains empty. TRENDBoard represents whether the trend of evidence in going from ADS-1 to ADS-2 reports is up or down and whether high value targets may be at risk from this launch event. The TrendCheck rule set uses numerical processing techniques to compare the infrared intensity, the sensor confidence and the sensor capability in both the DS-1 and ADS-2 report. In case of equal ratings, such as a sensor capability of FULL for both ADS-1 and ADS-2 the trend is held to be UP. Since there are 3 features which are compared there are three UP or DOWN votes. The number of UP votes determine the COMMENT and EXPNUMBER of TRENDBoard. Three UP votes result in an EXPNUMBER of 4 with the resulting comment: "TREND of evidence is: POSITIVELY UP." Lesser number of UP votes result in trends of UP, DOWN or POSTIVELY DOWN which also result in lower EXPNUMBERs. Having determined the trend, TargetCheck determines which high value (C3 node) targets are threatened by this event. Using the HEADING

given in the ADS-2 message, TargetCheck process the HIVALTAG property of the launch site actor. Recall that HIVALTAG takes the following form:

```
(LS7  [launch site actor property list]
   HIVALTAG   ((TARGET HEADING RANGE)   (TGT1 345 1047)
                 (TGT2 215 1756) and so on...)
```

Thus, the heading a missile from LS7 would have to take to attack TGT1 would be 345 degrees which lies at a distance of 1,047 kilometers from the launch site. Whereas BSS site detection involves finding a heading that lies in a range between two points, TargetCheck must determine whether a heading represents a threat to targets which lie at a single heading from the launch site. TRENDBoard shares the *name* of the instance variables with DETECTBoard, but they represent different things. Thus, *WILLDETECT* for a target represents those targets that lie within 1 degree of the missile heading. For example, if the ADS-2 HEADING is 214 degrees and the HIVALTAG shown above is processed then TGT2 would be on the WILLDETECT list. CLOSEDETECT is the list of targets that are within 5 degrees of the heading and thus within the maneuvering range of the TYPE2 missile, while FARDETECT are those within 10 degrees. A target is only listed on each list once, so that a TGT that appears on the WILLDETECT list is not listed on FARDETECT or CLOSEDETECT as well. The information obtained by TargetCheck is represented by a HIVALNUMBER, whose value is 1 through 4 and represents the threat presented to the C3 network by this missile heading. TargetCheck also appends the COMMENT found in TRENDBoard by

adding in the case where the WILLDETECT, CLOSEDETECT and FARDETECT lists are empty: "THREAT TO C3 Nodes: VERY LOW. " That particular COMMENT would have a HIVALNUMBER of 1, with other threat comments resulting in LOW, HIGH or VERY HIGH with a correspondingly higher HIVALNUMBER to represent the situation. Therefore, the TRENDBoard is now completed and its instance variables take the following form.

WILLDETECT	[Those targets within 1 degree of missile heading]
WILLNUMBER	[the number of targets appearing in WILLDETECT list]
CLOSEDETECT	[those targets within 5 degrees of missile heading]
CLOSENUMBER	[the number of targets appearing in CLOSEDETECT]
FARDETECT	[those targets within 10 degrees of missile heading]
FARNUMBER	[number of targets appearing in FARDETECT list]
HIVALNUMBER	[represents the state of threat to C3 network]
EXPNUMBER	[represents the trend of evidence from ADS-1 to ADS-2]
COMMENT	[first explanation layer, tied to EXPNUMBER]

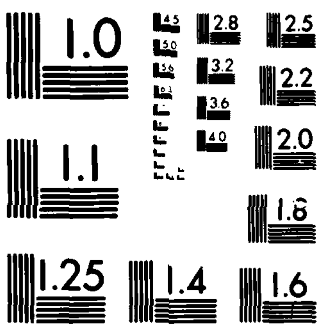
Filling out the TRENDBoard brings the process of specialist subprograms posting hypotheses to a close, which was triggered by the successful match of an ADS-2 report. SENTINEL now has five completed blackboards and an ADS2Confidence level to continue processing this report. Up to this point, the processing has been linear since all events regardless of their confidence level or supporting evidence have been processed in a very similar manner. Now, the ADS-2 confidence level will be the key to different processing for events.

Confidence Processing

Following ADS-2 matching there is now an EVENT property list which contains the ADS-1 and ADS-2 feature sets. Included in that EVENT, is the P2CONFIDENCE or second pass confidence, which will

AD-A163 827 AN EXPERT SYSTEM DECISION AID FOR A COMMAND CONTROL AND COMMUNICATIONS OPERATOR(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D L TOBAT 2/4

UNCLASSIFIED DEC 85 AFIT/GE/ENG/85D-50 F/G 17/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

determine all the further processing of the ADS-2 report. P2CONFIDENCE falls into five levels: HIGH, EXTHIGH, MIXED, LOW and EXTLOW. Rather than discussing the processing of each confidence level separately, an "ith" confidence level processing is discussed and then each case is differentiated. The ith confidence level processing is shown in figure 18, and represents all the functions that are invoked by any confidence level. The first rule set consulted by any of the confidence levels is BoardVoting. BoardVoting examines the EXPNUMBER, the number which summarizes the state of each of the five boards, and a vote is taken. Voting has been used frequently in sensor interpretation as a means of increasing the certainty of information obtained from sensor systems by combining the information from many sensors into a single composite sensor (12:161). The majority of the five filled blackboards will either confirm or deny the confidence level under consideration. For example, when the P2CONFIDENCE is initially HIGH, yet a majority of the blackboards are at a LOW and offer little evidence to suggest that an actual missile event is under way, the confidence level is reset to LOW. Not all five blackboards will always be filled. If no INT reports have been received since SENTINEL initialization, INTELBoard will have no information and thus no comment or EXPNUMBER on INTELBoard. When there is no discernible heading on the second pass report, there will be no information on DETECTBoard either. In these cases, further factors will be considered to maintain the voting procedure with an odd number of voters.

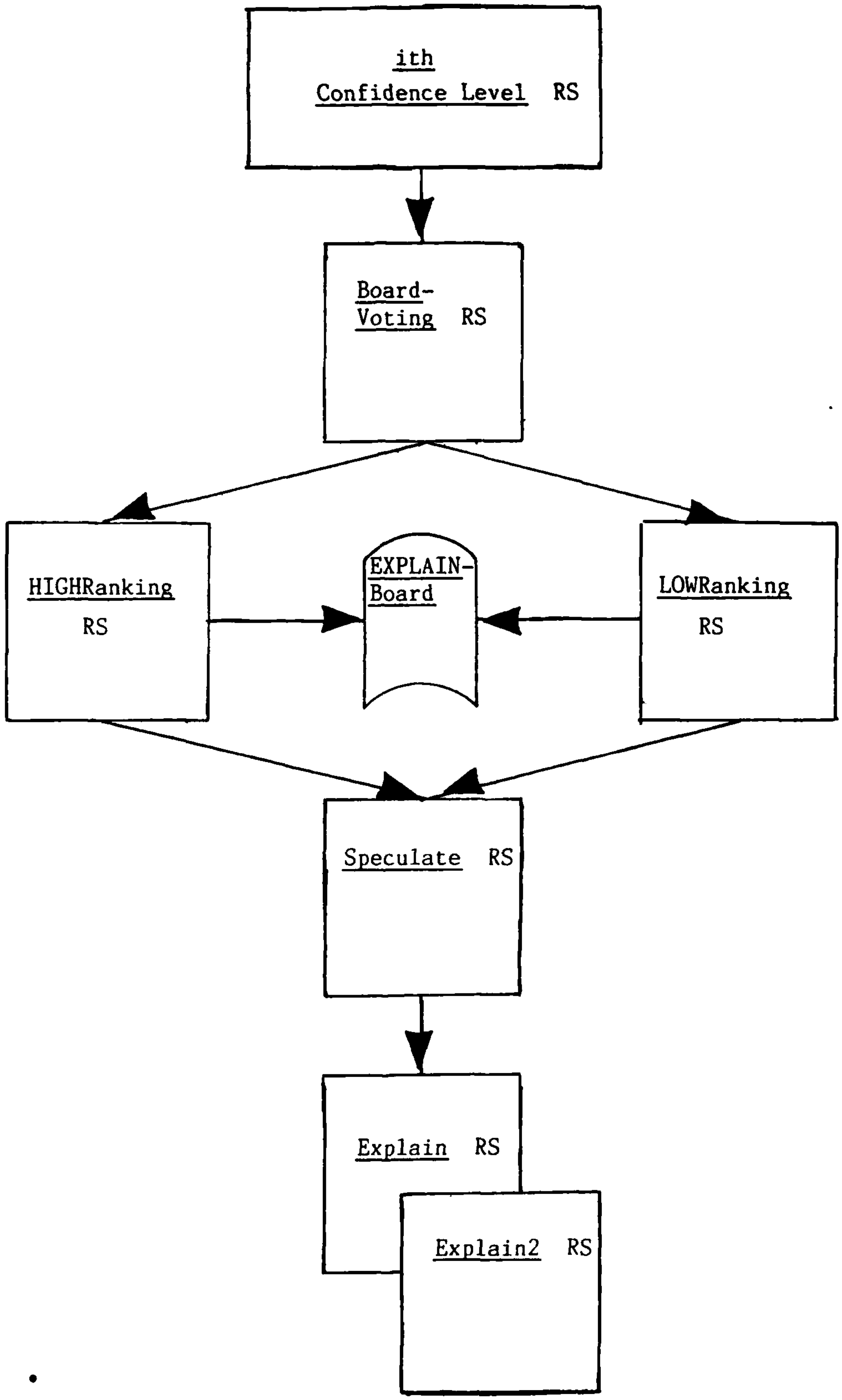


Figure 18. The ith Confidence Level Processing.

To enhance the logical order of presentation during explanation, the ith confidence calls on either HIGHRanking or LOWRanking to rank order the blackboards in the order that is most conducive to the ith confidence. Thus, if confidence is HIGH, HIGHRanking would be called to rank order the blackboards. HIGHRanking would examine the EXPNUMBER of each blackboard and rank them in order to have the blackboard which most clearly supports the HIGH confidence rating labeled as FACTOR1. The next one would be FACTOR2 and so on until all the five blackboards were ranked. If a blackboard did not have any information it would not be used as a FACTOR. In case of ties, where SYSBoard:EXPNUMBER is 4 and DETECTBoard:EXPNUMBER might also be 4, the precedence is decided by a FACTORLIST which is set inside each confidence level. Therefore, each confidence level can rank the evidence in a different priority, so that HIGHConfidence may list DETECTBoard first, followed by SYSBoard whereas LOWConfidence lists them in the reverse order. This is determined heuristically, since LOWConfidence would require confirmation that the sensor network is in good shape to detect any missile event and is not crippled; while for HIGHConfidence one can assume that the sensor system is functioning correctly and the most important factor is the detected missile heading. The FACTOR1 through FACTOR5 set by the HIGH and LOWRanking rule sets are stored in EXPLAINBoard, which is a blackboard whose function is to explain an event. Having completed all the instance variables that comprise the EXPLAINBoard, one can now see all the contents of EXPLAINBoard:

```
(EXPLAINBoard
  FACTOR1          [Blackboard which best supports the confidence
                     level being processed.]
  FACTOR2  FACTOR3     [Other blackboards in ranked order that are
  FACTOR4  FACTOR5     set in FACTORLIST and are HIGH or LOWRanking]
  COMMENT      [Comment from Speculate if special case triggered]
  EXPNUMBER    [indicates in numerical form the special case which
               triggered the COMMENT]  )
```

Following BoardVoting, the ith confidence level may call Speculate to determine if any special cases such as a first strike, widespread sabotage or a test condition either staged by THEM or perhaps erroneously by US. Speculate checks all the appropriate indicators and only returns a COMMENT and EXPNUMBER if one of the special conditions are triggered. Having confirmed the confidence level and run Speculate to determine if any special cases might be in effect, the ith confidence level can now be used to complete the ADS-2 message and explain its reasoning.

Having filled out the five factors in EXPLAINBoard with either HIGH or LOWRanking, the ith confidence level provides an explanation capability with two layers by calling on the Explain and Explain2 rulesets. Explain is invoked in response to MWO "WHY" queries and takes each FACTOR in ranked order from the EXPLAINBoard, and shows the abstract COMMENT associated with each board to the user. Explain2 provides the second layer of explanation, in response to "HOW" queries and represents the second level of explanation, that is the tabulated results of each block and message to the user. Thus, when the MWO requests an explanation, the COMMENT associated with the blackboard labeled as FACTOR1 which most supports the confidence level is shown to the MWO. The MWO can then ask another

WHY question, upon which he is shown the COMMENT from the FACTOR2 blackboard and so on. When the MWO asks HOW questions, the detailed data on each blackboard such as the contents of the SYSBoard are shown to the MWO. Once the explanation is completed, the ADS-2 and the ith level confidence processing is completed.

Differentiating the ith Confidence Level

There are five confidence level rule sets in SENTINEL, each one corresponding to the level of EXTHIGH, HIGH, MIXED, LOW and EXTLOW. MIXEDConfidence is the smallest and simplest rule set of the five. Basically, MIXEDConfidence tries to decide whether confidence level processing should be HIGH or LOW. At this point in SENTINEL processing there is enough information available to make a reasonable attempt to decide whether confidence is HIGH or LOW. MIXEDConfidence calls only on BoardVoting to decide which path to take. If there is no clearcut majority consensus by all the blackboards, the TRENDBoard will provide the required information. LOWConfidence is attempting to resolve a paradox; if there are enough indications for a second pass satellite message why is confidence LOW? LOWConfidence calls on BoardVoting to confirm the confidence level, which can be upgraded to HIGH if the number of UP votes are in the majority, while LOWRanking provides the ranking of the factors on EXPLAINBoard, and Speculate is invoked to determine if test conditions or unusual system outages are responsible for the ambiguous indications encountered so far.

EXTLOWConfidence differs from LOWConfidence by trying to get

the MWO to delete this event from the EVENTLIST. If the EXTLOW confidence is not verified by BoardVoting, the confidence level is returned to LOW. A transfer factor is set so that confidence levels can be changed only once, to preclude an EVENT which initially was ranked as EXTLOW, to be changed to LOW and perhaps even to HIGH otherwise. The premise behind EXTLOWConfidence is that a spurious report has occurred that does not represent an actual missile launch, and attempts to convince the MWO to place this EVENT into an inactive status by deleting it from the EVENTLIST. To do this, EXTLOWConfidence runs LOWRanking and then shows the user the abstract COMMENT of the three lowest factors as shown below:

SENTINEL DECISION AID CONFIDENCE is: EXTLOW

EXTLOW CONFIDENCE level confirmed. {Outcome of BoardVoting}

REPORTING SENSOR CHECK reveals CREDIBILITY of Sensor: VERY POOR {COMMENT from MATCHBoard}

Trend of Evidence is: POSITIVELY DOWN {COMMENT from TRENDBoard}

INTEL Reports: NO indications of ATTACK {INTELBoard COMMENT}

Based on the factors explained above, SENTINEL recommmends that this event be deleted from the active EVENTLIST. If you concur type T, otherwise type NIL to continue processing.

If the MWO selects NIL above, then EXTLOWConfidence processes the ADS-2 report in a manner quite similar to LOWConfidence. However, EXTLOWConfidence will not use Speculate to attempt to determine an explanation as LOWConfidence does because there will not be sufficient reliable information to speculate on. HIGHConfidence does not use Speculate either, since it assumes a threat is present unless proved otherwise. HIGHConfidence runs

through BoardVoting to verify the confidence level and then spends more time reasoning about what type of missile is present. Unless proved otherwise, the longer range TYPE2 missile with multiple warheads is assumed. HIGHConfidence is more concerned about the missile type since there appears to be ample evidence for an actual missile launch, and more processing time will be spent examining the HIVALTAG and the BSSTAG to verify the ADS-2 prediction of what BSS sites will detect this event and to predict what type of threat is facing the C3 network high value targets. Knowing the type of missile present gives SENTINEL the critical range information that along with the reputed missile heading can eliminate possible targets if they are situated beyond the maximum range of the missile.

EXTHIGHConfidence obviously represents an actual event that could potentially be a hostile threat, but it must be tempered by caution. For example a test launch by THEM could also result in EXTHIGH confidence or if a US sensor site was running exercise tests, an EXTHIGH confidence could result from a simulation that could be mistaken for an actual launch event. This was shown to be the cause of a recent NORAD false alarm that triggered public apprehension (21:27). Once its confidence level is verified by BoardVoting, EXTHIGHConfidence pays particular attention to Speculate as it examines the blackboards for evidence of a TEST, FIRSTRIKE or SABOTAGE event. SABOTAGE is a difficult state to reason with, since a "blinding" of the sensor system by sabotage would increase the possiblity of a hostile event, but this is offset by a

somewhat lower confidence level that could result from the damaged state of the sensor system. FIRSTSTRIKE indications are considered present when there is a large amount of evidence for a launch detection whose heading represents a threat to C3 nodes. Finally, TEST examines the SYS status to determine which US sites may be running tests on the system. EXTHIGHConfidence then calls on HIGHRanking to place the factors on EXPLAINBoard in a fashion suitable for explanation. At this point, the only thing lacking from filling out the ADS-2 report is a hypothesis. The MWO must decide whether the event represents a HOSTILE, TEST, UNKNOWN or UNRESOLVED event which is the subject of the next section.

Hypothesis Selection

To select a hypothesis, SENTINEL uses the concept of feature sets and distance from a prototype to decide what hypothesis the event most closely fits. This is a often used technique in pattern recognition cases discussed in the literature:

> Inherent in the processing of information acquisition is the process of pattern recognition. This process generally involves two phases: extraction and identification. A given stimulus is "coded" in terms of its features. The term "features" implies such characteristics as angles, lines, or edges. . . In the identification phase, the sensory-perceptual system classifies the stimulus object. The way in which this is often assumed to occur is by a weighted matching of the current feature list against a likely set of prototypes in long term memory with the input being classified according to the name of the best matching prototype (17:646).

SENTINEL is attempting to duplicate the human capabilities of pattern recognition, by providing a separate prototype for each

hypothesis as a rule set. These rule sets are weighted matching of the factors extracted from the event list and the filled blackboards. As shown in figure 19 there is a separate rule set for each hypothesis, which consist of the following; a HOSTILE event is one where the missile is directed towards the country of US. TEST events are those which have detected a missile launch whose trajectory will definitely pass outside the US, while UNRESOLVED is an event with a trajectory that may or may not be directed towards US. UNKNOWN events are those events where insufficient information is available to fit the other cases. To decide which hypothesis fits the event and thus identify the event, each confidence level examines the results of four rule sets as shown in figure 19. Each confidence level then selects a hypothesis based on a decision threshold that is particular to each confidence level.

HIGH and EXTHIGHConfidence are more concerned with differentiating a HOSTILE from a TEST hypothesis, while LOW and EXTLOWConfidence are more concerned with the distance between UNKNOWN and UNRESOLVED. Some crossover may occur however, so each confidence level does examine the distance to all four hypotheses. The four hypothesis rule sets are a "textbook example" or prototype of each type of event. HOSTILE for example represents a high probability of a launch coupled with intelligence indications warning of attack and a high threat to C3 nodes or US targets. TEST would have similar indications except that the threat to C3 nodes would be quite low and the possibilities of BSS detection would be marginal. Each of these hypotheses consider 12 factors, and of these

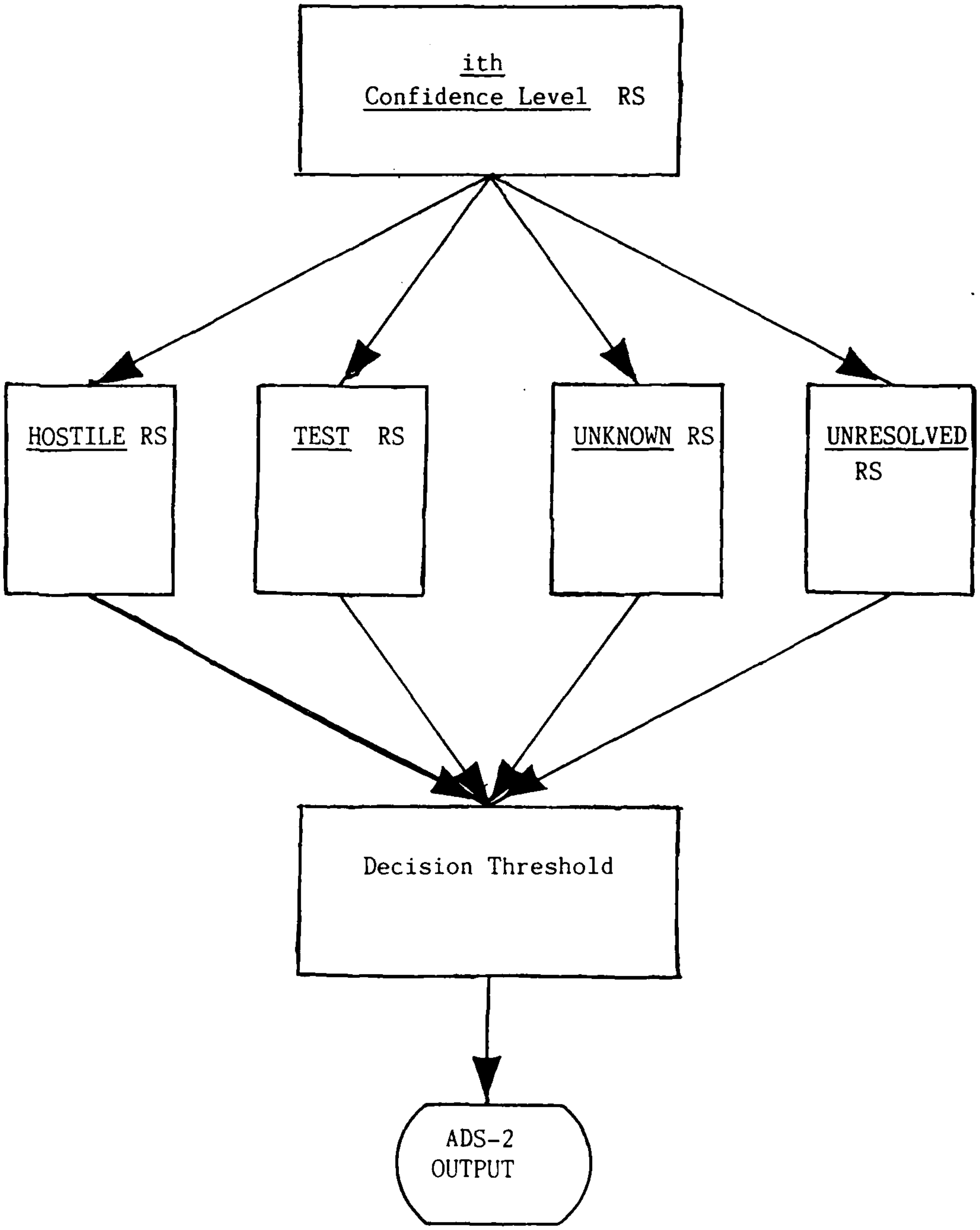


Figure 19. The ith Confidence Level Processing Continued to Show Hypothesis Selection.

12 factors, 2 of them are weighted more heavily than the others. By having equal number of factors and an equal number that are more heavily weighted, the four hypotheses can be said to be normalized. The two factors with additional weighting are the ones that are more crucial for hypothesis separation. For example, to differentiate TEST from HOSTILE, the possibilities of detection and the threat to C3 nodes are weighed more heavily. These heuristics have to be chosen carefully however. It would be tempting to just use the more heavily weighted factors to clearly separate all four hypotheses, but this is not possible because not all events result in the same amount of information. For example, if the heading is XXX or not discernible, which can be typical of an UNKNOWN event, the threat to C3 nodes and the possibilities of detection are NIL since without a heading no information is available. Restricting the factors to just detection and C3 threat information is not a valid approach. To separate UNKNOWN from UNRESOLVED, TrendBoard and the high value threat are used, and are weighted more heavily. It is interesting to note that the MWO using SIMCOPE does share the limitation of not being able to predict detection possibilities by using the SIMCOPE screen, since when the heading is XXX no direction of heading and consequently no FAN appears either. Therefore this limitation is not just a SENTINEL limitation but one central to SIMCOPE as well.

All four hypotheses are then run and the number of factors triggered by each are maintained in the variables: UNKNOWNNUM, TESTNUM, HOSTILENUM and UNRESOLVEDNUM. A variable called THREATNUM is also established by HOSTILE and it tabulates a definite

indication of THREAT by examining the DETECTBoard, the TRENDBoard and the high value threat. NOTHREATNUM looks at these factors plus the MATCHBoard to establish how much the threat can be discounted. When HOSTILE is invoked THREATNUM is 0 and is counted up as each threat is triggered by a rule while NOTHREATNUM is invoked with TEST or UNKNOWN. THREATNUM is a representation of how much belief that the event is a threat whereas NOTHREATNUM is a representation of how much evidence there is in the event to discount the threat. Having run thorugh all four hypotheses and come up with a value for each of the six variables, each confidence level can now select the appropriate hypothesis through a decision threshold.

As each hypothesis is run the one which has the most factors triggered is stored in HYPO1 while the second leading hypothesis in terms of factors triggered is called HYPO2. The other two hypotheses are discarded and SENTINEL is concerned only in the distance or the number of factors that distinguished HYPO1 from HYPO2. If the difference is 3 factors or more, the distance separation is held to be sufficient to pick HYPO1 outright. If the distance is less than 3, each separate confidence level uses heuristics to determine what the hypothesis is. In all cases a single hypothesis is selected. The HIGH and EXTHIGHConfidences assume that a threat is present unless NOTHREATNUM is high, and try to clearly separate the TEST and HOSTILE hypothesis. On the other hand, LOW and EXTLOWConfidence assume that there is no threat, unless THREATNUM is high and pay more attention to resolving the distance between UNKNOWN and UNRESOLVED events. In either case, this is a potential spot for

further developement and use of heuristics, to extend the size and separation of each feature set hypothesis and by extending the use of Speculate and other functions to not just determine what is happening, but also to determine exactly what further information based on the circumstances would be most beneficial to the MWO. At this point, SENTINEL is finally in a position to answer the ADS-2 report and explain its answers using two levels of explanation.

ADS-2 Report Output

At this point, the ADS-2 report will be filled out by SENTINEL in an identical manner to which the MWO fills out the SIMCOPE report. The user is then free to select further explanation and to examine either of the two levels of each blackboard. The filling out of the ADS-2 report is shown below:

```
      ADS  PASS 2    EVENT1        [the current EVENTMATCH]

1.  DETECTED: 142913      [PRESENTIME maintained by METASCHEDULER]
2.  ADS:  SOUTH           [the reporting satellite]
3.  EVENT TYPE: HOSTILE   [the hypothesis selected ]
4.  BSS DETECT:  6        [the BSS site that will detect the event]
5.  THREAT:  YES          [Based on THREATNUM and NOTHREATNUM]
6.  CONFIDENCE: HIGH      [confidence level processed]

WARNING! FIRST STRIKE INDICATIONS ARE PRESENT. [output of Speculate]

If you wish further explanation type T otherwise
type NIL to continue processing.
```

If the MWO selects option T above a short explanation of how the two levels of explanation are invoked and used. Recall that the first explanation capability is invoked by running the Explain ruleset and is triggered by WHY questions whereas the second, or

detailed layer of questions is invoked by using Explain2 and by answering HOW questions. The sample dialogue from SENTINEL below illustrates the process:

```
Explanation for ADS-2 processing is in two layers. The first
abstract layer can be accessed by typing in WHY followed by a
carriage return <CR>.
The second detailed layer is can be accessed by from the first layer
by typing in HOW <CR>. Typing NIL <CR> exits the program.

Type in WHY or NIL now:
[WHY]

Trend of Evidence is: POSITIVELY UP

Type in HOW for detailed explanation or WHY for further abstraction,
NIL to exit.
[HOW]

Confidence of the ADS-1 Message: HIGH
Confidence of the ADS-2 Message: HIGH

Reported Sensor CAPABILITY on the FIRST PASS is: PARTIAL
The Sensor CONFIDENCE on the FIRST PASS is: HIGH

The Reported SECOND PASS Sensor CAPABILITY is: FULL
SECOND PASS Sensor CONFIDENCE is: HIGH

The Infrared Intensity Reading reported on the FIRST PASS is: TYPE2
SECOND PASS IR Reading: IMPROVES over first pass reading.
SENTINEL concludes launcher type is: TYPE2

Type in WHY for more explanation, NIL to exit:
[WHY]

BSS Detection Opportunities are: VERY STRONG
Occuring at BSS sites: BSS4 BSS5 BSS6
Threat to C3 Nodes: VERY HIGH

Type in HOW to see details, WHY to continue, NIL to exit:
[NIL]
```

The order in which the explanation is given is set by the numbered factors on the EXPLAINBoard by using HIGHRanking or LOWRanking. At first, it was only necessary to run one or the other

ranking rule sets. For example HIGH and EXTHIGH always used HIGHRanking, while LOW and EXTLOW would use LOWRanking. In operation however, it was necessary to modify this strategy. It became obvious in practice that "context dependent" explanation was needed for logical consistency. When explaining a TEST event it was more consistent to explain what factors were low such as DETECTBoard racher than starting off by explaining all the "strong" factors which are quite similar to HOSTILE. This would result in confusion for the MWO since the explanation for a HOSTILE or TEST event would not usually be differentiated until the fourth or fifth factor was explained. In a similar fashion, HOSTILE is always explained by HIGHRanking since it is more essential to place the information in an order which logically places the "strong" factors supporting the HOSTILE hypothesis ahead of the lower numbered factors which are shown to the later to the MWO. A hypothesis of HOSTILE, no matter what the confidence level is always explained using HIGHRanking, while a TEST event regardless of the confidence level is ranked by LOWRanking to explain the output. This brings the ADS-2 processing to a close, and leaves only the remaining BSS radar site messages to be processed which is the subject of the next chapter.

V. Processing of the Third Pass Radar Report and Conclusion

The processing of the third pass BSS site report will use many of the concepts that were covered in the previous chapter. The blackboard model which led to the two layer explanation capability is reused by incorporating a few modifications. The BSS message is the report which describes the detection of launched objects at a BSS radar site. The BSS site notifies the MWO of the time of detection, the number of objects tracked, whether trajectory stability is present, along with a reported heading and a prediction for impact time of the object or objects. When more than one object has been detected, the BSS report will give the heading and impact time only for the center object. On the SIMCOPE screen, the MWO still sees a single fan even if multiple objects are present. And in the same manner as the ADS satellite report, the BSS site also issues a site confidence rating of 0 through 9. Having recieved a BSS report, SENTINEL must use some mechanism to decide how to match the new BSS report with an existing EVENT from the EVENTLIST.

Matching BSS Reports

As shown in figure 20, BSSHANDLER is the first function to process the BSS message, after METASCHEDULER has routed it. BSSHANDLER decomposes the BSS report into its constituent parts, and then attempts to match it. There are three steps that could result in a BSS match. In order to match BSS messages to the EVENT it was necessary to add P2HYPO, the determined ADS-2 hypothesis that was

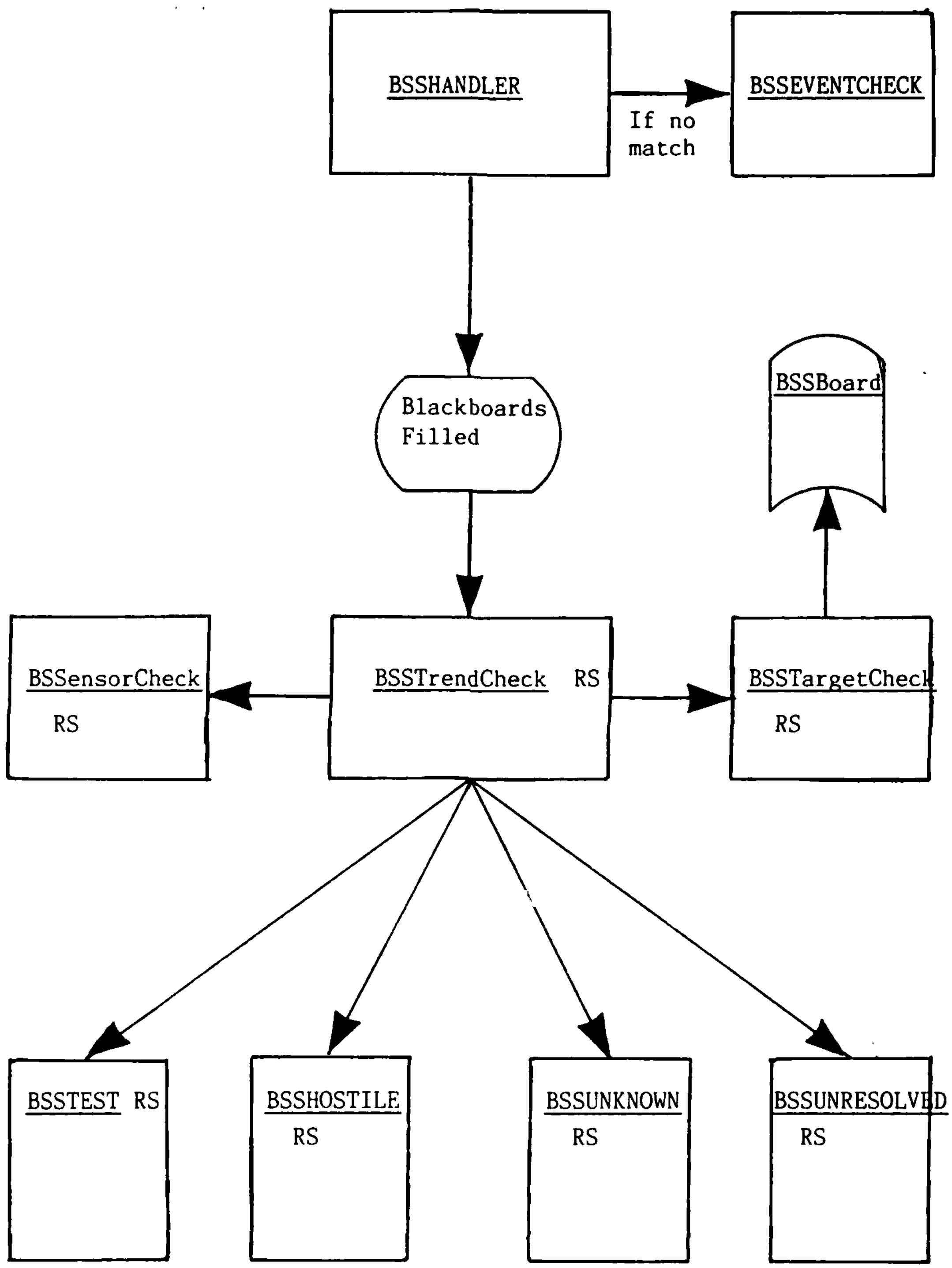


Figure 20. Initial BSS Message Processing That is Sent to a HYPOTHESIS Rule Set.

associated with this event, and P2HEADING to the EVENT property list. Before covering the automatic matching by BSSHANDLER, it is instructive to review what is now present in an EVENT property list after undergoing both ADS-1 and ADS-2 processing.

```
(EVENTNUMBER   P1LS [LAUNCHSITE it has been matched to, such as LS4]
    P1CONFIDENCE    [EXTHIGH, HIGH, MIXED, LOW or EXTLOW]
    P1SENSOR        [first pass numerical sensor rating]
    P1IRTYPE        [TYPE1, TYPE2, UNKNOWN or INDISTINCT]
    P2CONFIDENCE    [resulting ADS-2 confidence level]
    P2SENSOR        [second pass numerical sensor rating]
    P2TYPE          [MATCH, NOMATCH or MISMATCH]
    P2HYPO          [HOSTILE, TEST, UNKNOWN or UNRESOLVED]
    P2HEADING       [reported ADS-2 heading of object]
    P2TIME          [time at which BSS report expected]  )
```

The key to matching BSS reports to an EVENT will be matching the P2HEADING and the P2TIME to the object heading and time of report of the BSS message. However, the key to processing the remainder of the BSS report will be P2HYPO. By the time a BSS report has been received, the confidence level is no longer a major issue. Obviously, if two separate satellite reports and a radar site report have been received concerning a launch event, there can be little remaining doubt that an actual launch event has occured and is in progress. The key to further processing will be the whether the EVENT can still be interpreted as a HOSTILE or a TEST event. But first the BSS message must be matched to an EVENT.

BSSHANDLER will automatically match a BSS message to an EVENT based first on two criteria: the P2TIME must be within two minutes of the BSS detection time and, the BSS site would normally be able to detect this heading of an object launched from the P1LS of the EVENT in question. If BSS site #7 reported an object with a heading

of 207 and the P1LS of EVENT4 was LS6, the BSSTAG of LS6 could be checked to insure BSS coverage was compatible. If the BSSTAG of LS6 included this element, (BSS#7 134 167), EVENT4 could be removed as a candidate for BSS matching since the maximum heading from LS6 that could be detected by BSS site 7 would be 167 degrees, whereas the BSS heading is 207 degrees. For those events which can meet the first two criteria, the one whose P2HEADING is closest to the BSS heading will be announced as the EVENTMATCH. When no automatic match occurs, BSSHANDLER calls on BSSEVENTCHECK to complete the match if possible.

BSSEVENTCHECK first makes a "best guess" at what the EVENTMATCH is by relaxing the time consideration for up to 5 minutes deviation from P2TIME and by looking for the best combination of TIME and HEADING criteria. The "best guess" is announced to the MWO, who is shown the P2TIME and P2HEADING and the BSS time and heading. The MWO is free to accept this match or to look at the remaining EVENTs. For the remainder of the events, BSSEVENTCHECK checks through the EVENTLIST and presents to the MWO those EVENTs which have been through both ADS-1 and ADS-2 processing and are candidates for BSS matching as shown below:

```
Attempt to match: EVENT4

P2HEADING: 206                          P2TIME: 145600
HEADING reported by BSS: 207       BSS Detection Time: 145730

Range of coverage for this BSS site from the reported launch site
extends from 200 degrees to 189 degrees.

Type T to MATCH this event, NIL to see another attempt to match
or STOP to EXIT.
```

When a BSS match is made, P2TIME is deleted from the EVENT in question. Thus, an EVENT that has a NIL for P1TIME and P2TIME has gone through all 3 stages of SENTINEL processing, from ADS-1 and ADS-2 and through BSS. Recall that an EVENT with a value for P1TIME has only undergone ADS-1 processing. Having secured an EVENTMATCH for the BSS report, further processing can now occur that will be tied to the blackboard concepts used previously.

BSSBoard Processing

Having made a successful match, BSSHANDLER fills the blackboards in the same manner that ADS-2 processing did, by running the board filling rule sets such as SensorCheck, DetectCheck and TargetCheck. There are a few differences however, at the BSS level the output of the DETECTBoard which determined which BSS site could or could not detect is somewhat redundant. Therefore, BSSTargetCheck is run in place of HeadingCheck. The BSSTargetCheck rule set processes the FANCALL targets in the same manner that TargetCheck did for HIVALTAG during ADS-2 processing. The FANCALL targets are checked to see if they fall into three lists. The first list, WILLDETECT are for those targets that lie within 1 degree from the reported heading. CLOSEDETECT are those targets that lie within five degree of the object heading, and thus within manuvering range of a TYPE2 missile, while FARDETECT are those targets which lie up to 10 degrees away from the heading. There is also an associated NUMBER variable such as CLOSENUMBER, that records the number of targets in each list. A target can only be on one of these lists at a time.

These instance variables are recorded on the BSSBoard which is a blackboard solely for the purposes of BSS processing. Rather than write over the values of an unused blackboard at this level such as DETECTBoard, a new board was the most convenient and logical extension of the blackboard concept covered in the previous section.

In order to make the BSSBoard complete, a COMMENT and an EXPNUMBER must also be included. Accordingly, BSSTrendCheck uses the results of BSSTargetCheck to produce a BSSBoard COMMENT and associated EXPNUMBER. If the WILLNUMBER of BSSBoard is greater than zero, the COMMENT would be: "Threat to US targets: VERY LIKELY," and the EXPNUMBER would be 4 since this would be the state most indicative of a hostile event. In certain cases however, the threat to US targets could be quite LOW, while the threat to the high value targets could be quite HIGH which should result in a high EXPNUMBER. Thus, BSSTrendCheck also examines the threat to high value targets, which will override the EXPNUMBER for BSSBoard if necessary.

As its name implies, the BSSTrendCheck also checks the trend of evidence to observe whether it has increased or decreased from the ADS-1 to ADS-2 to BSS message. BSSTrendCheck uses BSSensorCheck to essentially compute a P3SENSOR rating which is not embedded in the EVENT. Instead the P3SENSOR rating can be compared with the P1SENSOR and the P2SENSOR rating to observe if the trend of evidence is rising or falling. BSSTrendCheck also checks the stability indication and the number of objects detected to see if it confirms or denies the previous P1IRTYPE and P2TYPE indications of what type of launcher is involved. A maneuvering warhead or one with multiple

launch objects would be indicative of a TYPE2 rather than a TYPE1 missile. Almost the only way to secure a DENIAL of missile launcher type would be an EVENT which had detected a TYPE1 missile on the first pass, achieved a MATCH or NOMATCH P2TYPE but yet had a multiple object or maneuvering detection on the BSS report. The COMMENT associated with TRENDBoard such as "Trend of Evidence: UP" will stay the same but will reflect the BSS message contribution. Having filled all the previous blackboards and filled the new BSSBoard, the only remaining step for BSSTrendCheck is to route the EVENTMATCH to the appropriate hypothesis processing rule set. By examining the P2HYPO of the EVENTMATCH, BSSTrendCheck will then invoke one of the four hypothesis rule sets.

Hypothesis Processing

The key to BSS processing is the value of P2HYPO of the EVENTMATCH, which can invoke one of the four hypothesis rule sets: BSSTEST, BSSHOSTILE, BSSUNRESOLVED and BSSUNKNOWN. Each one of them first revises the confidence level by invoking BoardVoting as was done previously for ADS-2 processing, and as shown in figure 21. If the number of HIGHVOTEs is large enough, the confidence level could be upgraded. These cases were kept to a minumum since confidence levels are not as vital for BSS processing as they were for ADS-2 processing. What is vital is the hypothesis, which can also be revised by the output of BSSBoard. If the hypothesis is TEST for example, but the EXPNUMBER on the BSSBoard is HIGH, indicating a serious threat to US targets, the hypothesis is set to HOSTILE and

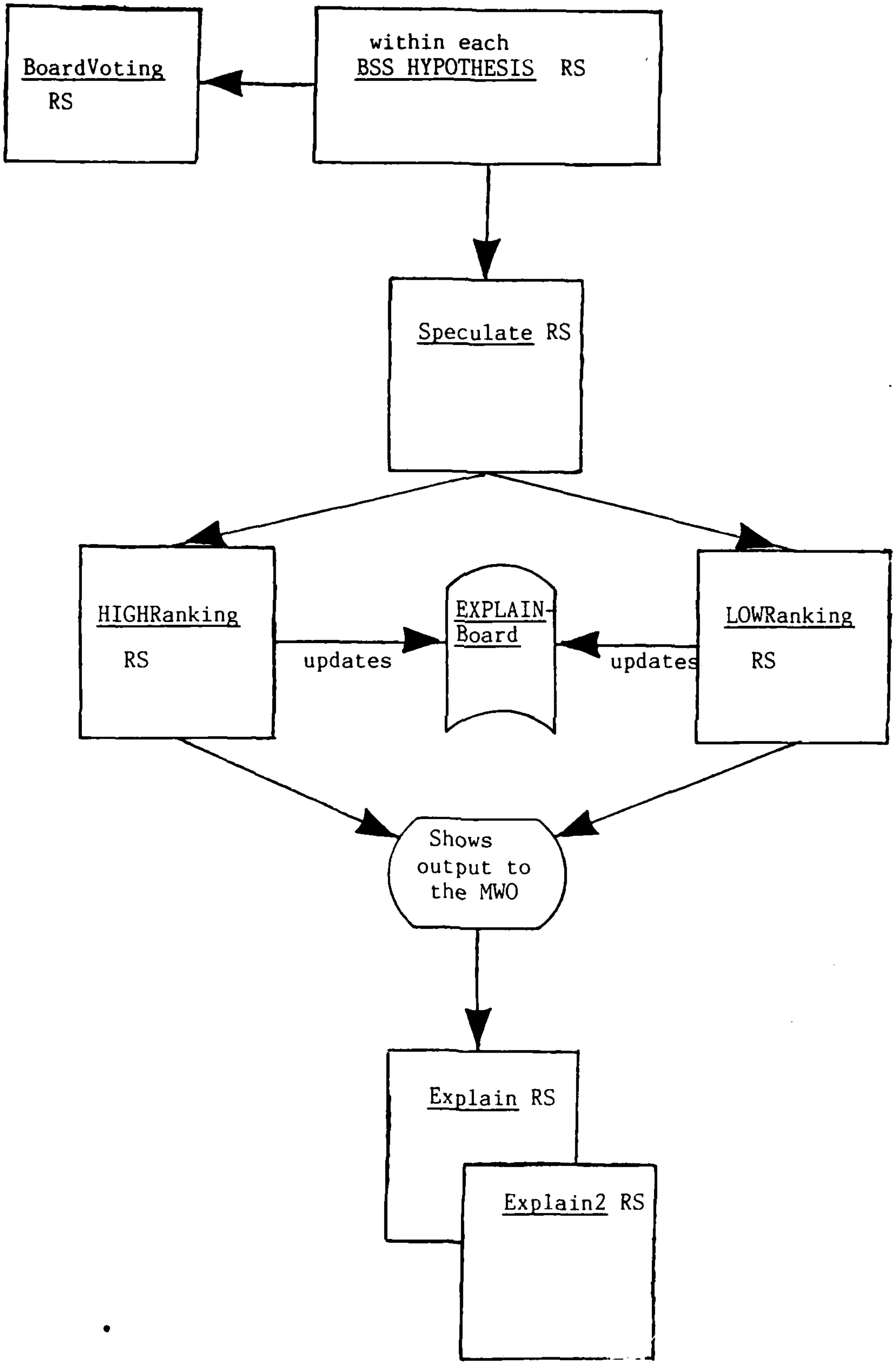


Figure 21. Completion of BSS Message Processing Within Each BSSHYPOTHESIS Rule Set.

BSSHOSTILE is run instead. A flag called HYPOTRANSFER insures that these hypothesis revisions can only occur once to avoid the problem of infinite loops that could occur with multiple hypothesis revisions. While BSSTEST assumes there is no threat unless proved otherwise, BSSHOSTILE assumes a threat. BSSUNRESOLVED and BSSUNKNOWN attempt to secure an explanation for the evidence present by using Speculate and other heuristics. Generally the pattern will be to force the hypothesis to either TEST or HOSTILE depending on the threat or lack of threat the launch event poses. Having confirmed the resulting hypothesis the BSS report can be answered for the MWO, as in the following output:

```
EVENT considered:  HOSTILE        [from BSSHOSTILE]

     BSS    EVENT1                [EVENT1 is the EVENTMATCH]

1. DETECTED: 143317
2. BSS: 6
3. THREAT: YES
4. LAUNCHER TYPE: TYPE2
5. OBJECTS TRACKED: 1
6. VERY LIKELY TARGETS: (HAYES, LIDO)    [BSSBoard:WILLDETECT]
7. EST IMPACT TIME: 143538
8. CONFIDENCE: EXTHIGH

If you wish further explanation type T, otherwise type
NIL to proceed.
```

Item 6 which describes the threat to US targets varies according to which BSSBoard list is not empty. A filled WILLDETECT list results in the line above, while a CLOSEDETECT would list "LIKELY TARGETS" and a FARDETECT would result in "POSSIBLE TARGETS" while if all these lists are empty result in the line "6. POSSIBLE TARGETS: NONE." Once the MWO is shown the BSS report output, the two layer

explanation capability described for ADS-2 processing is also replicated at this level. Context dependent reasoning also occurs at the BSS level, so that a HOSTILE event always uses HIGHRanking to rank order the blackboard into the explanation order listed in EXPLAINBoard. When no threat is indicated LOWRanking is used to set the FACTORs in the order which would most clearly explain to the operator why no threat indications are present. As before the explanation capability is provided by the Explain and Explain2 rule sets, where Explain answers the "WHY" queries and Explain2 responds to the operator's "HOW" questions. Both explanation rule sets were modified to incoporate the BSS processing. The only major changes were to have the trend of evidence second layer explanation include the BSS site confidence and capability along with the determination that: "BSS report" CONFIRMS or DENIES "the indicated missile type." Because DETECTBoard at the BSS level of processing does not offer much information, BSSBoard was substituted for DETECTBoard in the order of explanation. Thus, to explain a BSS report, one of the FACTORs on the EXPLAINBoard will include BSSBoard but not DETECTBoard. To illustrate these changes, assume that the operator has requested further explanation for the BSS report shown above. What the MWO could see for the explanation of the BSSBoard would be the following:

```
THREAT to US Targets: HIGHLY LIKELY
THREAT to C3 Nodes:  HIGH

Type in WHY to see further comments, HOW to see details
of the above or NIL to exit.
[HOW]
```

```
The following TARGETS lie within 1 degree of the
reported heading: (RIVERTON  NORTHCITY  DELTA)
THREAT to these sites is: ALMOST CERTAIN.

The following TARGETS lie within 10 degrees of the
reported heading: (PINEGROVE  OCENACITY)
THREAT to these sites is: POSSIBLE.

These C3 sites lie within 5 degrees of the
reported heading:  BRF
THREAT to these sites is: LIKELY.

Type in WHY to see further comments, HOW to see details
of the above or NIL to exit.
[NIL]
```

Finishing the BSS message explanation brings the processing for that particular EVENT to a close since all three types of reports, ADS-1, ADS-2 and BSS messages have been matched. At this point the EVENT is deleted from the EVENTLIST since it is no longer an active EVENT and is placed in the EXPIREDEVENTLIST. This makes the remainder of SENTINEL processing computationally easier since each attempt at a MATCH involves examining each EVENT on the EVENTLIST. To demonstrate the goal of computational feasibility, ten worst case trials of an EVENT, which resulted in a previously unknown launch site, were tracked to completion. Bear in mind that the time listed below is processing time, and excludes the time spent in functions which require extensive user inputs such as the explanation capability.

ADS-1: 31.4 seconds   ADS-2: 9.6 seconds   BSS: 8.3 seconds

The above numbers are not an average, they are the longest time it took for any of the ten events. The relatively large amount of time spent processing the ADS-1 message is due to the polar representation computations. By contrast the worst case times for

EVENTs which resulted in automatic matching, and thus required no polar computations was:

ADS-1: 1.6 seconds   ADS-2: 5.8 seconds   BSS: 5.6 seconds

For a system that has not been optimized for speed, this demonstrates computational feasibility. Use of a larger or faster processor would result in faster response times, but this could be negated by adding more targets, launch sites and processing features to SENTINEL. Having brought SENTINEL processing to a close, it is now appropriate to reflect on the conclusions and some recommendations for further research.

Conclusions

The devlopment of SENTINEL brings to mind some observations that could be useful for further expert system development of decision aids. One must first determine the appropriate physical representation of the system under consideration. By examining all the information available, some method of representation should emerge as a logical candidate. In the case of SENTINEL, the availability of only headings and geographical coordinates made a polar representation a likely and logical way of expressing the SIMCOPE world. The decision aid must also have full access to all the information available to the operator it is attempting to help. When the MWO looks at the SIMCOPE graphical display, he is obtaining a great deal of information that SENTINEL must also be able to "know." By using the SCENARIO REPRESENTATION, SENTINEL is also able to "know" what the MWO is seeing on the display. The use of a

forward chaining, rule based hypothesis appears to be appropriate for SENTINEL as it was for the sensor interpreting expert systems which suggested the architecture. An important factor in being able to complete SENTINEL was the fact that SIMCOPE had already quantified and abstracted the task of the MWO. In this manner, not much time had to be spent finding a "domain expert" and understanding the domain first. It would be an interesting study to observe if expert system building could be done more effectively if the "domain expert" was asked only to quantify and abstract his task independently of the interview process and brought in to interrelate with the expert system builder only in the testing and refinement stage. The use of expert system building tool software was also quite pivotal in assuring the completion of SENTINEL.

The use of numerical methods for processing helped minimize the number of rules used and the computational time required. These numerical methods were an easier way of combining three or more factors into a single figure of merit, and were suggested by the theory of endorsements by Paul R. Cohen. Cohen's theory was a source for several ideas used in SENTINEL, such as the ranking of hypotheses and using numerical methods of combining evidence. ADS-1 evidence could be compared and ranked in SENTINEL against the ADS-2 or BSS data for a launch event. The use of the pattern recognition concept of feature sets and prototypes were the key to selecting a hypothesis and were a more conventional way of dealing with uncertainity in an expert system without resorting to "certainty factors" or to somewhat flawed Bayesian analysis. The remainder of

SENTINEL is dominated by the idea of using abstaction levels to explain its output.

Explanation capabilities of expert systems have traditionally consisted of showing the invoked rules which led to a particular conclusion or citing references. Neither of these techniques would have been satisfactory for SENTINEL. By recognizing that four levels of information exist: the SIMCOPE messages, property lists of the actors, contents of the blackboards, and the abstract comments associated with each blackboard, SENTINEL could offer a two layer explanation capability. The abstract comments are not quite precise but they serve a useful purpose in showing a readily understandable message. Indeed, the output of a phrase such as: "SENSOR SYSTEM STATUS: POOR. Leading Cause: SABOTAGE" gets a lot of information across to an overworked operator. However, only offering abstract comments is not a complete solution, since operators will not trust a system that cannot offer a more detailed and accurate explanation for such a comment as the second layer of SENTINEL explanation does. And such an explanation must be context dependent. Providing abstract comments in a random fashion is probably worse than not providing any at all. Only by using context dependent explanations can the abstract comments be ranked in a fashion which supports the hypothesis in the same manner that a debater might try to win an an argument. It is important to note that SENTINEL will present all the evidence whether it supports the hypothesis or not; it is only the order of presentation that is modified and not the explanation itself. SENTINEL is also indebted to the concept of blackboard model

expert systems and the use of object oriented programming for the development of an explanation capability. Although SENTINEL has been completed, there are ways in which it could be profitably extended as detailed below.

Recommendations

The major area of improvement for SENTINEL would be in its interaction with the user. Being a decision aid this aspect of SENTINEL is the most important for its effective use. The method of SENTINEL interaction should be examined to see in which way it could be most effective. Using simply a mouse or touch screen to indicate a reponse would be one step. Using the keyboard as is done now is the logical initial approach but is probably one of the worse ways for the MWO to interface with SENTINEL. Use of more natural language inputs would be a good starting point, but eventually the use of voice commands or a touch screen would be the best solution. To really test and evaluate the combination of SENTINEL and SIMCOPE, it is first necessary to interconnect the PDP-11 to the Xerox 1108 and permit SENTINEL to receive the SIMCOPE messages directly. Rather than having the SENTINEL output on a completely separate screen, it would also be ideal if the output of SENTINEL could appear directly on a small portion of the SIMCOPE screen or perhaps on a closely adjacent monitor. Once this was accomplished it will be possible for the AAMRL to evaluate exactly how effective of a decision aid SENTINEL could be. The stress levels and performance of MWO operators with no exposure to SENTINEL could be compared to those

who were taught to use it at the same time they learned of SIMCOPE. In this manner the effectiveness of such a decision aid could be established. SENTINEL could also be tailored to each operator's tastes as some may always prefer a certain type of explanation or output format. In actual use such a system could know certain things about the operator such as his experience level and prior preferences and tailor its performance accordingly. In respect to its rule sets, SENTINEL could be further extended through the use of more rule sets such as Speculate. The eventual goal would be to assist the MWO further by recording his pattern of inquiry, and thus actively help the MWO to find the specific data which he most needs at the time. SENTINEL would certainly be an excellent starting point for further C3 expert system decision aid development in almost any domain.

Bibliography

1. AlphaScience. SIMCOPE Instructions. Dayton, Ohio: February 1984.

2. Bachant, Judith and John McDermott. "R1 Revisited: Four Years in the Trenches." The AI Magazine, 5: 21-32 (Fall 1984).

3. Bobrow, Daniel G. and Mark Stefik. The LOOPS Manual. Palo Alto, California: Xerox Corporation, 1983.

4. Cohen, Paul R., Edward A. Feigenbaum and Avron Barr, editors. The Handbook of Artificial Intelligence. Stanford: HeurisTech Press, 1982.

5. Cohen, Paul R. Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach. Boston: Pittman Publishing Incorporated, 1985.

6. Cohen, Paul R. and Milton R. Grinberg. "A Theory of Heuristic Reasoning About Uncertainty," The AI Magazine, 3: 17-24 (Summer 1983).

7. Davis, Randall. "Expert Systems: Where are we? And where do we go from here?," The AI Magazine, 3: 3-22 (Spring 1982).

8. Dillard R.A. and R.J. Bechtel. Computing Confidences in Rule-Based Systems. San Diego, California: Naval Ocean Systems Center FY-82 Annual Report, 1982.

9. Duda, Richard O. and John G. Gaschnig. "Knowledge-Based Expert Systems Come of Age," Byte Magazine, 6: 258-262 (Sept 1981).

10. Feigenbaum, Edward A. and Pamela McCorduck. The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World. Reading, Massachusetts: Addison-Wesley Publishing Company, 1983.

11. Forester, John., Human Engineering Division. Personal Interview. Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, 26 July 1985.

12. Fox, Mark S. and others. "Techniques for Sensor-Based Diagnosis," Proceedings of the Eighth International Joint Conference on Artificial Intelligence, I: 158-163 (August 1983).

13. Funk, Ken and others. Prototype Intelligent Tactical Assistant (ITA). San Diego, California: Navy Personnel Research and Development Center, Report Number TR 84-38, May 1984.

14. Hayes-Roth, Frederick and others. Building Expert Systems. Reading, Massachusets: Addison-Wesley Publishing Company, 1983.

15. Klein, Gary A. "Automated Aids for the Proficient Decision Maker," Proceedings 1980 IEEE Conference of the Cybernetics Society, I: 301-304 (October 1980).

16. Lenat, Douglas B. and others. "An Expert System for Indications and Warning Analysis," Proceedings of the Eighth International Joint Conference on Artificial Intelligence, I: 259-262 (August 1983).

17. Masui, Shoichi and others. "Decision Making in Time-Critical Situations," Proceedings of the Eighth International Joint Conference on Artificial Intelligence, I: 233-235 (August 1983).

18. Rich, Elaine. Artificial Intelligence. New York: McGraw-Hill Book Company, 1983.

19. Sage, Andrew P. "Behavioral and Organizational Considerations in the Design of Information Systems and Processes for Planning and Decision Support," IEEE Transactions on Systems, Man and Cybernetics, 11: 640-678 (9 September 1981).

20. Stefik, Mark and others. "Knowledge Programming in LOOPS: Report on an Experimental Course," The AI Magazine, 4: 3466-356 (Fall 1983).

21. U.S News and World Report. "Nuclear War by Accident- Is it Impossible?" 95: 27-28 (19 December 1983).

22. Wallich, Paul. "Technology '84: Software," IEEE Spectrum, 21: 47-49 (January 1984).

23. Winston, Patrick H. Artificial Intelligence. Reading, Massachusets: Addison-Wesley Publishing Company, 1984.

Appendix A

A Basic Description of Expert Systems

Expert systems are specialized computer programs which are receiving considerable attention in scientific, industrial, and defense circles. Expert systems are part of the field of Artificial Intelligence which has been defined as "The study of how to make computers do things at which, at the moment, people are better" (18:1). Within this flexible definition, Artificial Intelligence currently covers such subjects as converting natural language automatically into computer language and helping robots to see. To date, the most successful and widespread use of Artificial Intelligence has been through expert systems. Expert systems, also known as knowledge based systems, are software programs which are designed to represent and apply specialized knowledge in a narrow field of expertise so that users can solve problems which previously could only be done by a specialist. According to Frederick Hayes-Roth, expert systems differ from other Artificial Intelligence tasks by doing the following:

1). Perform difficult tasks at expert levels of performance.

2). Emphasize domain-specific problem solving strategies.

3). Employ self-knowledge to reason about their own inference process and provide explanations or justifications for the conclusions reached.

4). Fall into several categories: interpretation, prediction, design, planning, monitoring and instruction (14:5).

The task of constructing expert systems usually begins by finding a domain expert and learning from him what rules of thumb or heuristics he uses to solve problems. These heuristics, along with specialized domain knowledge are transformed into IF-THEN rules. A control structure which may group rules into seperate categories in a hierarchical or a nonhierarchical fashion is usually used. In a conventional expert system, the IF-THEN rules are termed the "knowledge base" and the control structure which determines how the rules are sequenced is called the "inference engine." The most popular expert system architecture is the rule based production system which contains both knowledge sources and control structures. Other approaches include the use of "blackboards" which post data in an area accessible to all subprograms. Expert systems must also deal with many of the fundamental issues that are raised below:

1). Knowledge representation: There are numerous ways to represent the knowledge in a particular domain. Such techniques as frames, models and object oriented programming are the most common. Objects represent information by combining instructions and data into an entity which communicates with other objects by sending messages. Frames can be used to represent the concept of inheritance. For example; If the default value for the COLOR variable of a brick is RED, then one knows that the color of BRICK27 is RED unless specific information attached to BRICK27 indicates otherwise.

2). Search: With complex knowledge bases of thousands of rules, matching a specific rule in such a large search space becomes a very

time consuming task. Therefore, limiting the search or searching in a more efficient manner insures computational feasibility.

3). Dealing with Uncertainty: Most complex real problems and the work of most domain experts entail dealing with uncertainty. Many rules are not absolute but have a probablity associated with them. Most rule based systems deal with uncertainty by using "certainty factors" which are a subjective assesment of how much truth can be associated with a given fact. Other methods include Bayesian statistical methods which are more in line with conventional probability theory, and heuristic methods.

4). Explanation Capability: Most people will only trust an expert systems which can explain how it arrived at a conclusion. Explanation capabilities vary greatly, and can help verify the knowledge base in the testing phase. One popular method for explanation is the recital of all the rules that were invoked to lead to a conclusion.

These issues can be explored further by describing some examples of actual expert systems which have been used in fields where there are shortages of human experts. Three diverse systems which have been in use for over five years are MYCIN, a medical diagnosing aid; XCON which configures computers; and PROSPECTOR, which looks for oil and mineral desposits.

MYCIN (Shortliffe, 1976) addresses the problems of diagnosing and treating infectious blood diseases. Because laboratory tests that identify the specific organism involved in these swift and deadly diseases can take weeks to run, it is essential to start

treatment as soon as a reasonable attempt at diagnosis can be made. The attending physician enters a dialogue with MYCIN by supplying the symptoms and available test results and MYCIN then replies with its diagnosis and a indication of certainty, and can even answer some questions on how it arrived at its conclusion. MYCIN's skill at diagnosis is reported to be at the level of human specialists in infectious diseases and sometimes far above the level of nonspecialist physicians (10:65). The major emphasis in MYCIN is the explanation capability which traces the system's inferences and cites literature sources relevant to its conclusion. MYCIN also reasons with uncertainty by using certainty factors and uses a hierarchical control structure to direct its inquiries (14:53).

Another successful system is the XCON (McDermott, 1982) or R1 system used by the Digital Equipment Corporation to configure customer requests for the VAX family of minicomputers. By automatically laying out equipment cabinets and calling out the appropriate type and quantity of power supplies, disk drives and other minor items, XCON permits the configuration of VAX computers by sales personnel within minutes instead of burdening technical experts with tedious and time consuming work (14:40). XCON started its operational life in 1981 with 850 rules and by 1984 had over 3,250 rules, demonstrating the startingly large growth of the knowledge base and the capabilities of an expert system over a period of time. XCON is also distinguished by using "contexts" to separate its large knowledge base into natural groupings of rules. In this manner, the search problem is constrained to only the

average 100 rules or so in a context group instead of searching over all the rules each time (2:22).

PROSPECTOR (Duda, 1979) was developed for the French oil company Elf Acquitance, and assists in mineral exploration by analyzing the underlying rock structure for mineral or oil deposits. PROSPECTOR has already more than paid for itself by discovering a molybdenum deposit whose ultimate worth will exceed $100 million (14:6). PROSPECTOR contains rule based models for different types of ore deposits that were developed based upon interviews with leading geologists. PROSPECTOR asks the user yes or no and numerical input questions that must be acompanied by the user's confidence in the answer on a -5 to +5 scale; where -5 is a certain no, 0 indicates no knowledge about the certainty and +5 is a certain yes. In response, PROSPECTOR can suggest which tests would be most valuable to run next, and give the rationale for it, as well as provide informal education about each class of deposits (9:260).

The cornerstone of an expert system is its knowledge base, which is represented by the use of IF-THEN production rules. The top statement in figure 22 is a simple example of an IF-THEN rule from the MYCIN expert system, which illustrates what conditions are necessary to support a certain conclusion. The second production rule is a more complex version of an IF-THEN statement from the XCON system. The numbers present in the PROSPECTOR and MYCIN rules represent certainty factors. Expert systems are able to explain their reasoning process and acquire new knowledge and modify the old by changing their IF-THEN rules appropriately (18:288).

MYCIN

If: the stain of the organism is gram-positive, and
the morphology of the organism is coccus, and
the growth conformation of the organism is clumps
then: (0.7) the identity of the organism is staphylococcus.

R1

If: the most current active context is distributing
massbus devices, and
there is a single-port disk drive that has not been
assigned to a massbus, and
there are no unassigned dual-port disk drives, and
the number of devices that each massbus should
support is known, and
there is a massbus that has been assigned at least
one disk drive and that should support additional
disk drives,
and the type of cable needed to connect the disk drive
to the previous device on the massbus is known
then: assign the disk drive to the massbus.

PROSPECTOR

If: magnetite or pyrite in disseminated or veinlet form is
present
then: (2, -4) there is favorable mineralization and texture
for the propylitic stage.

Figure 22. Three Examples of IF-THEN Production Rules (16:286)

By their very nature, expert systems are currently limited to fields in which the rules are not contradictory and where the problems they are dealing with are well bounded and do not depend common sense. Hayes-Roth summarizes the shortcomings of expert systems by pointing out:

1). They are unable to recognize or deal with problems for which their own knowledge is inapplicable or insufficient.
2). They have no independent means of checking whether their conclusions are reasonable.
3). Explication of their reasoning process is frequently silent on fundamental issues (14:55).

Besides these fundamental limitations, widespread use of expert

systems has been hampered by the computing resources they require and by the amount of time it takes to build them.

The easiest way to program IF-THEN statements and use them in a language that resembles natural English is through the use of a computer language called LISP. LISP programs utilize a great deal of computer memory and are slowed by the great number of rules that must be processed (14:9). Very fast processing of expert systems software in LISP can be performed by dedicated computers available for $30,000 to $70,000 from such vendors as Symbolics and Xerox (22:47). Though some applications have been hampered by the price of these LISP machines, the major constraint is the amount of time necessary to produce an expert system.

Even with the increasing number of expert systems in use, building one is still a very time consuming task. In the words of Randall Davis of MIT, "Even for the best understood problems, experienced researchers using the best understood technologies still require at least five manyears to develop a system that begins to be robust" (7:10). To reduce this amount of effort and permit the use of less experienced programmers, extensive software tools such as Xerox's LOOPS package have been developed. LOOPS and other expert system building tools are optimized for the rapid building of rule bases, the use of frames to permit inheritance and can use object oriented programming. More specific information on the use of LOOPS can be found in Appendix C.

The greater use of programming aids and the growing capabilities of LISP machines will lead to the greater use of expert

systems in the military. One system that has been used is the Navy's HASP/SIAP system which interprets ocean sounds received by a passive sonar surveillance system (10:65). Rather than performing expensive signal processing on all the received data, the HASP/SIAP system sifts through it first, making allowances for local weather conditions and intelligence reports, and returning only the data it cannot explain for further processing. In coming years, expert systems can be expected to play an increasing role in such applications as battlefield control for commanders, the pilot associate program, and interpretation of sensor and intelligence data.

Appendix B

Outline of SIMCOPE Operations

The only fully documented record of the Simulated C3 Operator Performance Evaluation (SIMCOPE) is the SIMCOPE instruction manual which is listed as reference 1. The intent of this section is an attempt to give the reader the flavor of what the SIMCOPE Missile Warning Officer (MWO) operator is actually asked to do, and the specific format of SIMCOPE messages, without going into as much detail as the instruction manual. Specific descriptions of the scenario map used in SIMCOPE as well as the usual sequence of events are discussed at the beginning of Chapter II and are not repeated here. Instead, the general objectives of SIMCOPE are discussed first, followed by the SIMCOPE MWO operator's task and a detailed description of how each type of SIMCOPE message can be interpreted by using tables.

There are five types of messages used in SIMCOPE; the INT intelligence report, the SYS system status report, the ADS-1 and ADS-2 first and second pass satellite reports and finally the BSS radar site report. The basic objective of SIMCOPE is to function as a test bed for the C3 operator task of the strategic Missile Warning Officer (MWO). Specifically, SIMCOPE is to model the task of the MWO and evaluate different ways of presenting information and asks the MWO to assess the degree of mental effort required.

SIMCOPE Objectives

SIMCOPE is a computer simulation written in FORTRAN and running on a PDP-11 at the Aerospace Medical Research Laboratory at Wright-Patterson Air Force Base, Ohio. SIMCOPE is similar to a war game and is being used to study human information processing and decisionmaking in a realistic setting. SIMCOPE uses a fictional scenario and a model of the world that uses 20 countries. The task of the MWO is to watch and evaluate events and determine which if any launch events represent a possible threat. The objectives of SIMCOPE are to observe the accuracy and speed with which the MWO handles large amounts of information and completes and sends event reports (1:1).

The Subjective Workload Assessment Technique is used as a way of capturing the indicated workload level of the MWO. Upon completion of message processing, the MWO is asked for his subjective evaluation of whether he was pressed for time as well as the mental effort and psychological stress generated by that particular sequence and speed of messages. This measurement technique provides a way of representing the quality of the decisions made by the MWO, and the speed with which such reports can be handled (1:51). SIMCOPE was developed by several members of the Human Engineering Division of the Aerospace Medical Research Laboratory in 1983.

SIMCOPE Touch Screen Display

To perform his task, the MWO uses a workstation with a touch screen display as shown in figure 23. The large display area at the center of the screen is where the map showing the countries of US and THEM appear. At the top right hand side of the screen, one can see the readiness state, and the present or mission time. The blank area beneath the touch buttons labeled "MODE CONTROL" is where several buttons are located which permit various features of the map to be surpressed. The normal mode for SIMCOPE is "AUTO," which permits certain features to be done automatically by the computer. The actual SIMCOPE messages appear in the "MESSAGE DISPLAY AREA." The "EVENT KEYS" permit the assigning of an ADS-1, ADS-2 and BSS report to an event. These three types of messages announce themselves to the operator by sounding an alarm and by causing the "ACKNOWL" touch key, just above the scroll keys in figure 23, to flash. The MWO then presses the ACKNOWL key to stop its flashing, which cause the "ASSIGN" key which is located to the immediate right of the "ACKNOWL" key to begin flashing. By pressing the "ASSIGN" button and then pressing one of the "EVENT KEYS," labeled "E1" through "E8," the MWO assigns that particular message to an event.

The "DISPLAY KEYS" shown at the bottom of the screen permit past messages such as SYS and INT reports to be displayed in a variety of modes. Lastly, the "SCROLL KEYS" permit unassigned messages that are stacked in the message display area to be moved up or down. The

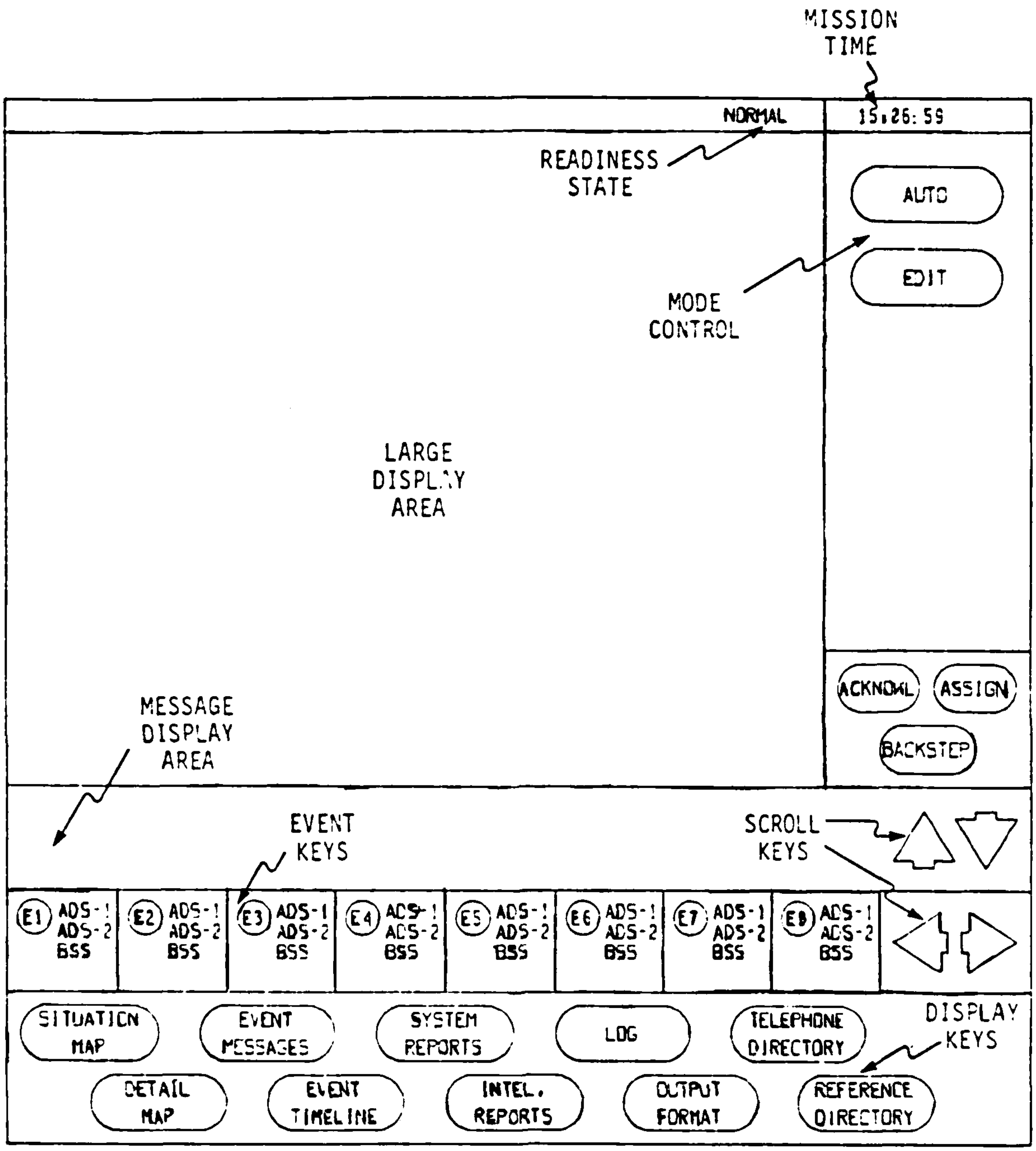


Figure 23. Representation of the SIMCOPE Touchscreen and Major Display Areas (1:8).

sideways scroll keys permit event keys to be scrolled; pressing the left button for example, would bring events labeled "E9,""E10" and so on to be displayed on the screen. Not all of the keys are shown in figure 23, figure 24 shows what appears on the SIMCOPE touch screen when the "DETAIL MAP" display code is pressed. Note that the display area is now dominated by the map of US and THEM. Note also to the right of the map that several more buttons have appeared in the "MODE CONTROL" regions. These permit the MWO to surpress some of the features displayed on the map such as the military installations or the ADS satellite or BSS radar zone of coverage. One can appreciate that the SIMCOPE screen becomes very cluttered when various missile launches and trajectories are displayed on the map as well. Having discussed the workstation the MWO uses, it is now appropriate to see the format for each type of message.

INT and SYS messages

The INT and SYS SIMCOPE messages are the simplest type to process for the MWO since no response is required. These messages arrive in the message display area announced only by a flashing "ACKNOWL" key. The MWO merely presses the "ACKNOWL" button to stop the flashing and the INT or SYS message leaves the display area. It is important to note that the INT and SYS reports are stored in a chronological listing that can be summoned at any time by pressing the "INTEL REPORTS" or "SYS REPORTS" keys located in the bottom of

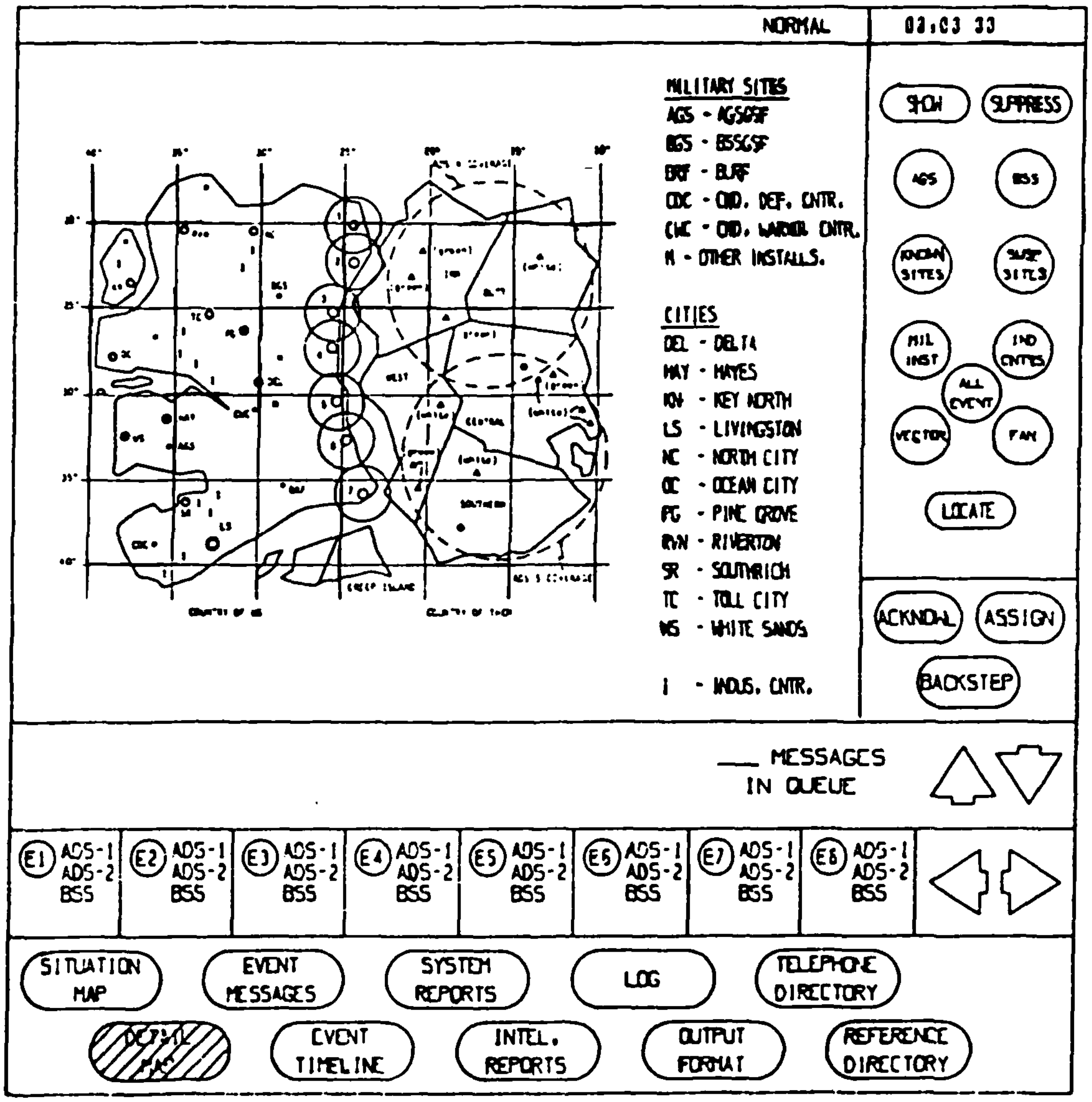


Figure 24. Detail Map Display(1:98).

the screen. As more messages arrive, the screen scrolls and the older messages are deleted from this listing (1:108). Although the messages stored in the listing are in "translated" form such as "The BSS site #6 sensor is now at PARTIAL capability due to SABOTAGE," the incoming SIMCOPE messages are a collection of alphanumeric characters separated by slashes, such as:

//CWC/INT//13/-7/5/143100/000000/+13/XXXXXX//.

Table 1 on page 146 demonstrates how this combination of a "raw" SIMCOPE INT message can be broken down into the appropriate data fields which represent different factors. The reader may recall that in SENTINEL the message format was altered to have parantheses instead of slashes for the ease of LISP manipulation. The SIMCOPE message shown at the top of table 1 is distinguished by having INT as the second element of the message. Recall that INT messages inform the MWO of news reports of the rest of the world that could affect the judgement of the MWO. The only field of the INT report which was not used by SENTINEL is field number 5 which represents the casualty levels incurred as a result of some reported action. This portion was not used since the Intelligence Model developed for this project would not use casualty levels incurred by other countries.

SYS messages are intended to notify the MWO of equipment operating conditions. SYS messages classified as "Routine" result in the same procedure as the INT messages. Mainly, the operator only

has to select the flashing "ACKNOWL" key to acknowledge the report. No other action is expected. An "emergency" SYS report will also cause an alarm to sound when received. The MOW is expected to examine the messages, note carefully what it means and when the "ACKNOWL" messages is pushed, the SYS messages leaves the message display area for storage in the SYS chronological listing (1:14). Details for decoding the SYS messages can be found in table 2, which shows the data fields which constitute the SYS report. All of these data fields are used in SENTINEL to update the sensor actor or are posted on the SYSBoard blackboard. Having discussed the INT and SYS messages, it is now appropriate to discuss the message processing that is tied to a launch event report.

ADS-1 Message Processing

When the MWO receives an ADS-1 report, the "ACKNOWL" key flashes continously and an alarm sounds. The MWO should then press the "ACKNOWL" key, after which the "ASSIGN" key begins flashing and the prompt "SELECT EVENT NUMBER" appears in the lower right corner of the display. Once the MWO selects an event number key to assign this ADS-1 report, the message disappears from the screen to be stored in a event timeline display which is accessed by pressing the "EVENT TIMELINE" key in the display key area. At this point, the MWO must fill out an ADS-1 report form which is shown in figure 25. Note that the key captioned "OUTPUT FORMAT" has been pressed in figure

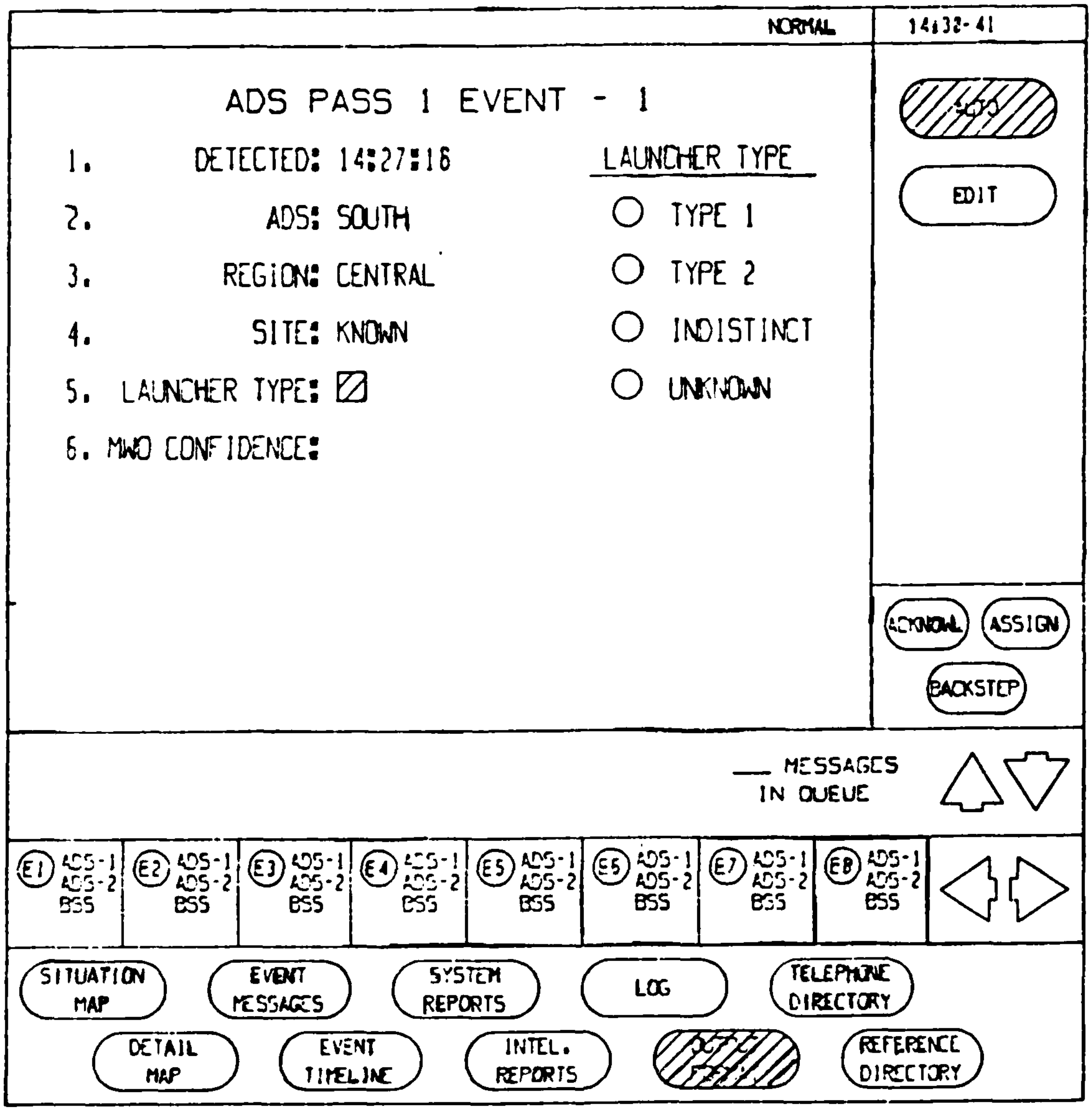


Figure 25. ADS-1 Display Following Launch Site Specification: Launcher Type Selection (1:24).

25, and the location of the other display functions such as "EVENT TIMELINE" as well. As shown on the screen of figure 25, a form entitled "ADS PASS 1 EVENT -1" announces the type of message and which event it has been linked to. The MWO must fill out the 6 items listed on the left. When SIMCOPE is an AUTO mode the first two items; DETECTED and ADS are filled out automatically. For the remainder of the items the MWO selects an item from a menu on the touch screen. As shown in figure 25, the current item the MWO is assigning is the LAUNCHER TYPE and he can select a response from the four item menu listed on the right hand side of the screen. Once he selects an item, SIMCOPE fills it in on the screen and proceeds to the next item. The completed form of an ADS-1 report can be seen in figure 26.

When all six items of the ADS 1 report have been filled, the MWO must now send the message to his higher headquarters, the Command Defense Center. The Release Authority number is used to simulate an authentication procedure to permit the CDC to verify that the MWO indeed originated this message. All the other items of this message are straight forward with the exception of the MWO CONFIDENCE which is item 6. The MWO CONFIDENCE is a subjective assessment of whether an event is to be considered hostile or not, and is assigned on a scale of 0 to 9 where 0 indicates a very high confidence that the event is not hostile (1:19). In a similar fashion to INT and SYS messages, the ADS-1 report can also be decomposed into several data

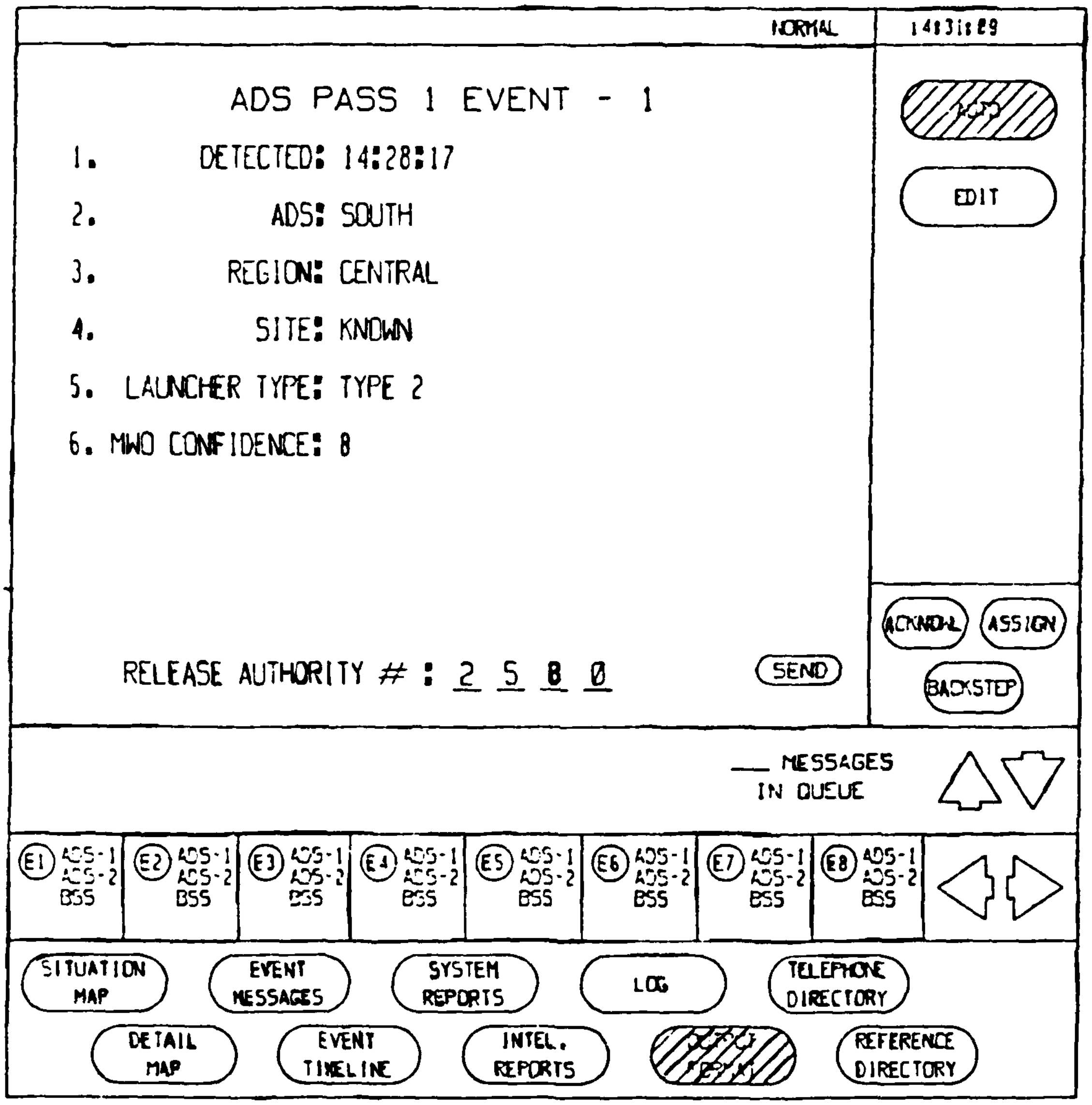


Figure 26. ADS-1 Display after Entering Release Authorization Number (1:27).

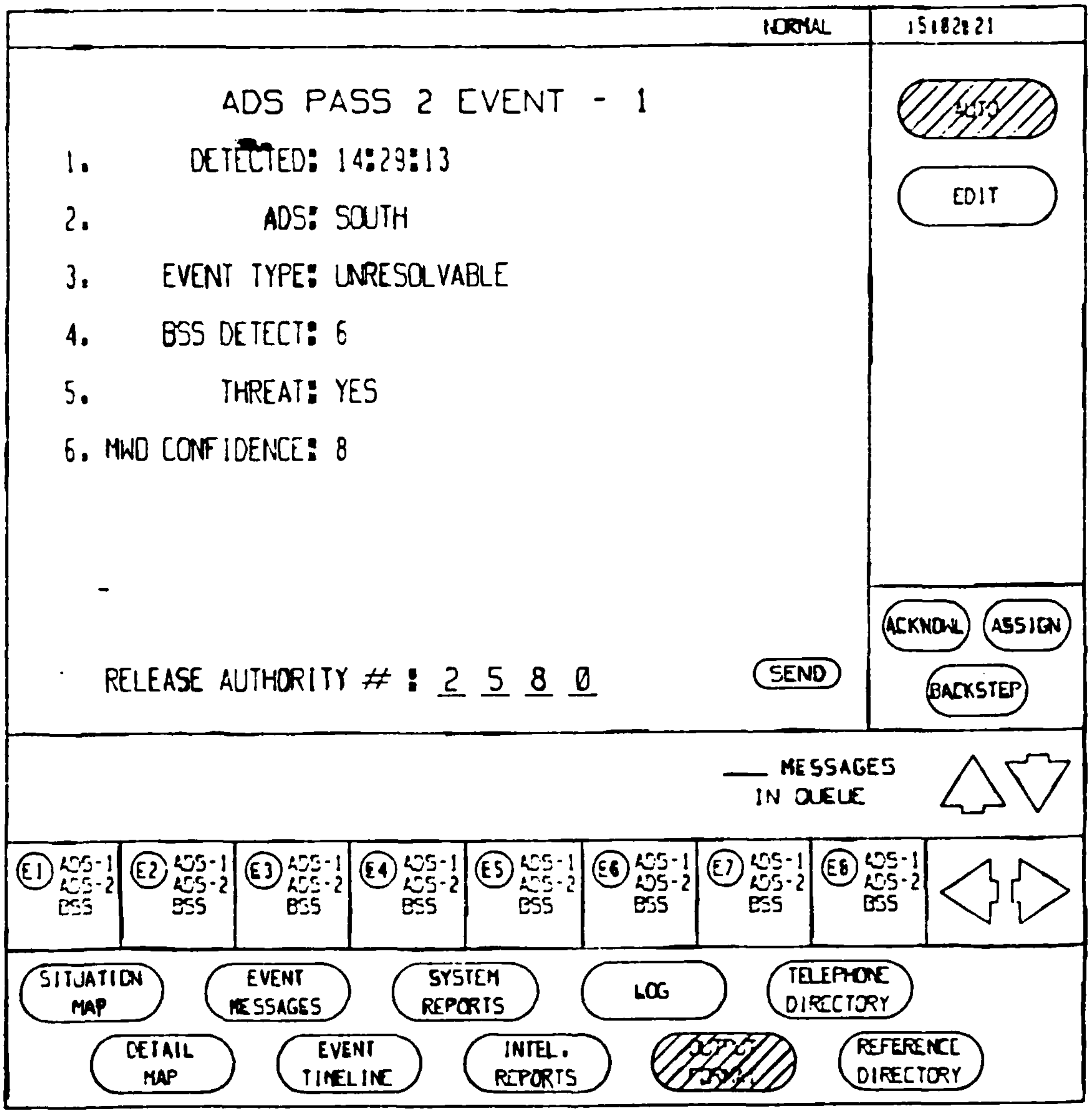


Figure 27. ADS-2 Display after Entering Release Authority Number (1:39).

fields as shown in table 3. All of the items shown in table 3 such as infrared intensity strength, reported launch latitude and longitude and the sensor site confidence indicating whether this data is valid are all used by SENTINEL. The only remaining messages are the ADS-2 and BSS site messages which are processed in a fashion very similar to the ADS-1 report.

ADS-2 and BSS Message Processing

ADS-2 messages announce themselves and are assigned to an event in an identical fashion to the way ADS-1 messages where described above. Because more information is available on the second satellite pass the response form to an ADS-2 event is quite different from the ADS-1 case and a completed form is shown in figure 27. When SIMCOPE is running in the AUTO mode, the first two items the DETECTED time and the ADS satellite are filled out automatically. The EVENT TYPE item asks the MWO to decide whether the event under consideration represents a HOSTILE, TEST, UNRESOLVABLE or UNKNOWN event. As described by the SIMCOPE instruction manual, HOSTILE events are those in which the reported trajectory of the missile falls within the country of US, while TEST events are those missile trajectories that lie wholly outside the US. Those events which result in a missile fan which is partially in US are held to be UNRESOLVED, while those for which the data is inconsistent are held to be UNKNOWN (1:76). The fourth item, BSS DETECT is the number of the BSS

site which is most likely to detect this missile, while the THREAT requires a simple YES or NO response. The MWO CONFIDENCE and RELEASE AUTHORITY NUMBER are filled out just like they were for ADS-1. The ADS-2 messages are interpreted using the data fields described in table 4.

BSS messages are also announced by a flashing "ACKNOWL" key and an alarm. Upon matching them to an event number, the MWO is also required to fill out a BSS report. The completed BSS report is shown in figure 28, where the first four items; DETECTED time, BSS site number, THREAT yes or no and OBJECTS TRACKED are automatically filled in. The MWO must select the response for POSSIBLE TARGETS using a menu that displays all the targets labeled by two or three letter designations. General military and industrial targets in the country of US are not identified in this fashion although cities, and C3 nodes are. EST IMPACT TIME is also filled in automatically once the MWO selects a possible target. Finally MWO CONFIDENCE and RELEASE AUTHORITY NUMBER are filled in the same manner ADS-1 and ADS-2 messages did, and the message is sent to the CDC. The data fields that constitute the BSS event message are shown in table 5. The most important characteristics of the BSS report are the number of objects being tracked, the expected time of impact and the trajectory stability which indicates whether a stable ballistic trajectory has been followed by this particular missile.

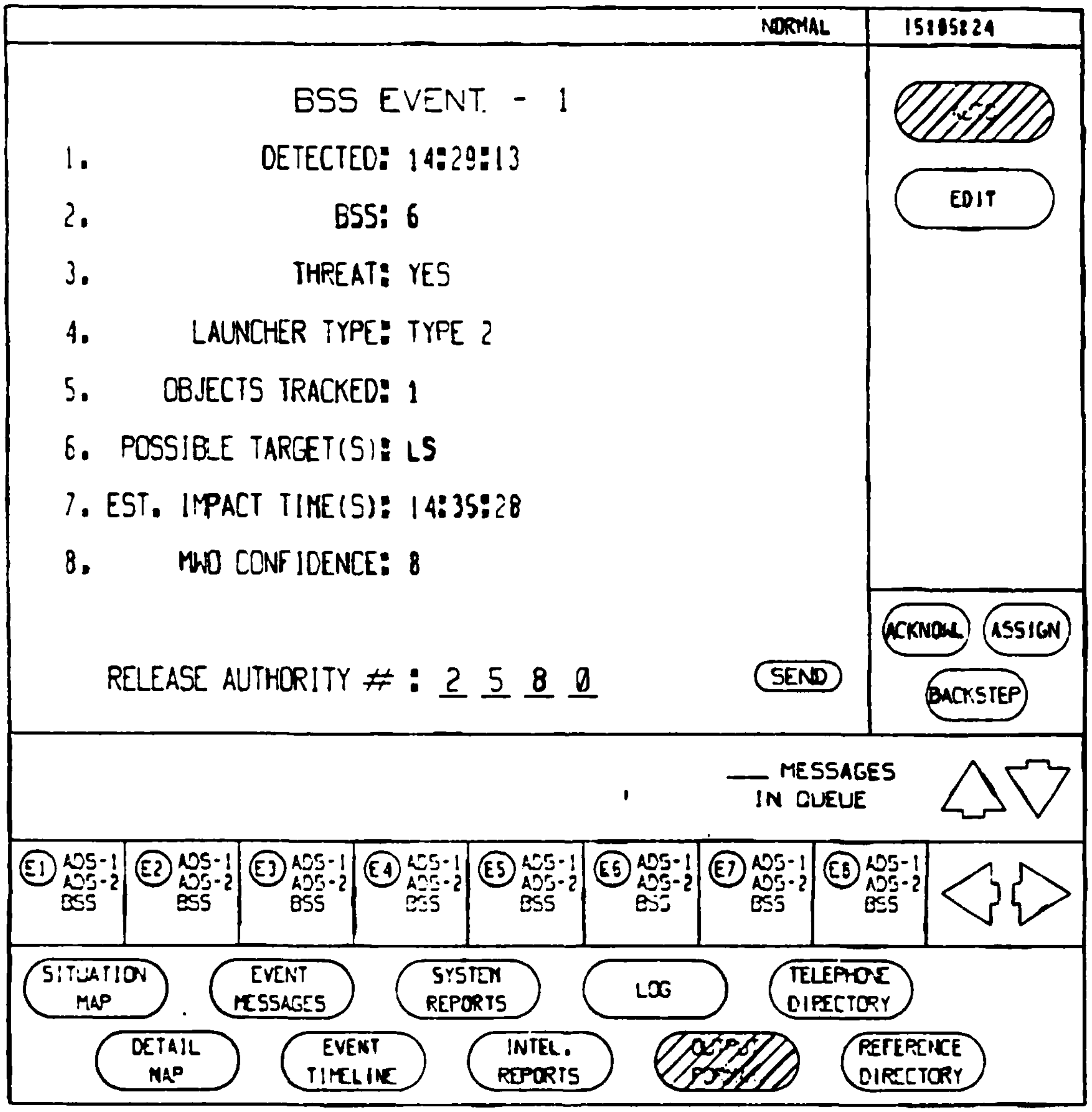


Figure 28. BSS Display after Entering Release Authority Number (1:48).

Conclusion

The SIMCOPE instruction manual is quite detailed and describes the situations that fall into each category using a type of check list. How SENTINEL determines these categories is well discussed in the main text and will not be repeated here. Instead, the task of the MWO and the types of messages he must deal with have been discussed to give the reader an appreciation for the task for which SENTINEL is intended as a Decision Aid. SIMCOPE is just a simulation, but one that can yield valuble insights into the task and operation of the C3 operator that functions as an MWO.

Finally, table 6 is a listing of all the targets and launch sites used by SENTINEL. The PROPERTY TAG represents the actual name for the property list under which the rest of this information is stored and accessed. Where SIMCOPE provided a name, the author used the SIMCOPE name. Latitude and longitude was determined by using the detail map display shown in the SIMCOPE instruction manual and are accurate to within half a degree (1:3). The designations fell in the follwing categories for targets; a city was a non-C3 node which had an alphabetic SIMCOPE designation, a C3 site was one of the C3 nodes that comprise the missile warning network, MILITARY or INDUSTRIAL targets are labeled M or I in the SIMCOPE map and were named by the author. For launch sites the types are as follows: GREEN is a suspected site, RED is a known site, RED1 is a known site that has TYPE1 missiles and RED2 is a known site that has TYPE2 missiles.

TABLE 1 (1:63)

THE INT MESSAGE STRING

EXAMPLE: //CWC/INT//13/-7/5/143100/000000/+13/XXXXXX//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
1. Who was responsible for the action? Country Name Identified by Number	0 - UNIDENTIFIED OR UNKNOWN 1 - ANYBODY 2 - ANYONE 3 - ANYPLACE 4 - BACKLAND 5 - BOTTOMLAND 6 - EASTLAND 7 - INLAND 8 - NOBODY 9 - NOPLACE 10 - NORTHLAND 11 - OUTBACK 12 - OVERTHERE 13 - SOMEBODY 14 - SOMEONE 15 - SOMEPLACE 16 - SOUTHLAND 17 - THEM 18 - TOPLAND 19 - US 20 - WESTLAND	The country of Somebody did...
2. Did (or Will Do/ Is Expected to Do) What?	+ - Has - - Has Not 1 - Threatened Invasion 2 - Threatened Attack 3 - Invaded 4 - Attacked 5 - Defeated 6 - Surrendered 7 - Allied 8 - Test Launch	Has not created an alliance with... (or dissolved one if it had previously existed)
3. To (or with) Whom?	0-20 (as in 1 above) Exception: If (1) was 19 and (2) was ± 7, then (3) cannot be 17	The country of Bottomland. (complete message: The country of Bottomland is no longer and ally of Somebody.

TABLE 1 (Cont'd)

THE INT MESSAGE STRING

EXAMPLE: //CWC/INT//13/-7/5/143100/000000/+13/XXXXXX//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
4. When? (expected time when action identified by field 2 will occur)	Time or date. Time in hours, minutes, and seconds (hhmmss): 000001 to 240000. A date as: ±01-01 to 12-31 XX-YY or UU-UU Time values specify same day; otherwise date is implied; a positive numeric value is day and month of this year; minus sign means last year; XX-YY, or UU-UU means either the date is more than a year away or is unknown.	The alliance was dissolved at 1431 ZULU today.
5. With What Result? If action (field 2) was 3 or 4, then what damage was inflicted? Level Codes: 0 - None 1 - Low 2 - Moderate 3 - Heavy 4 - Severe	First Digit: Military Target Destruction Level Second Digit: Industrial Target Destruction Level Third Digit: Civilian Target Destruction Level Fourth Digit: Military Casualty Level Fifth Digit: Industrial Casualty Level Sixth Digit: Civilian Casualty Level	The dissolution of the alliance had no target destruction or casualty implications

TABLE 1 (Cont'd)

THE INT MESSAGE STRING

EXAMPLE: //CWC/INT//13/-7/5/143100/000000/+13/XXXXXX//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
6. Reported By	+ Open Source - Intelligence Source 0-20 (as in 1 above) Exceptions: a) if open source, 19 is not applicable. b) if intelligence source, 17 is not applicable.	The Country of Somebody openly announced the dissolution of their alliance with the Country of Bottomland
7. Correlated Message	Same coding as field 4 except time should precede (be earlier than) present message time.	There was no prior message correlated with this one (indicated by the string of six X's).

TABLE 2 (1:66)

THE SYS MESSAGE STRING FOR SYSTEM STATUS REPORTS

EXAMPLE: //CWC/SYS//R/2/S/T/130523/150000/F//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
1. Type of System Status Message	R - Routine E - Emergency	System outage was planned.
2. System Affected	A: ADSGSF B: BSSGSF C: CDC F: BURF N: ADS North S: ADS South 1-7: BSS Site No.	BSS Site 2 is the affected site.
3. Affected Subsystem	C: Communication D: Data Channel H: Data Link (high speed) L: Data Link (low speed) P: Processor S: Sensor T: Telephone (voice)	The sensor itself will be (or was) inoperative
4. Reason for Outage	T: Test M: Maintenance F: Failure S: Sabotage U: Unknown	The sensor outage was scheduled to permit some kind of test.
5. Time of Subsystem Outage	000001 to 240000 (will be XXXXXX if site originating the message does not know this value).	The sensor was shut down at 1300 hours 05 minutes, 23 seconds ZULU.

TABLE 2 (Cont'd)

THE SYS MESSAGE STRING FOR SYSTEM STATUS REPORTS

EXAMPLE: //CWC/SYS//R/2/S/T/130523/150000/F//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
6. Estimated Time of Return to Service	000001 to 240000 (will be XXXXXX if the site originating this message does not know)	If it's past 1500 hours ZULU, the site should be back "up" and operating again.
7. Current Capability	F: Full P: Partial Z: Zero (none)	This message indicates the site was restored to operation at 1500 hours ZULU.

TABLE 3 (1:69)

THE ADS EVENT MESSAGE STRING FOR PASS 1

EXAMPLE: ADS Pass 1 Message Set: //CWC/ADS//S1/142718/8/-29/+13/142915/8//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
1. Detection Source and Pass Number	N or S 1	South ADS Pass 1 Message
2. Time of Detection	hrmmss* 000001 to 240000	Sensor detection occurred at 1400 hours, 27 minutes and 18 seconds ZULU.
3. Detection Strength of Sensor Intensity	0 to 9 Intensity level of the signal as detected by this sensor; zero is low and 9 is high.	Sensor intensity is 8, relatively high, higher than for a Type 2 launcher.
4. Launch Latitude	-15 to -45 (15°S to 45°S)	29° South, about the middle of Them.
5. Launch Longitude	+10 to +24 (10°W to 22°W)	13° West, about the middle of the CENTRAL region.
6. Expected Time of Arrival (ETA) for the ADS-2 Message	hrmmss 000001 to 240000	Expected time of arrival for the Pass 2 message, expressed in ZULU hours, minutes and seconds: 1400 hours, 29 minutes, 15 seconds ZULU.
7. Sensor Site Confidence	0 to 9 The sensor site's confidence that these data are valid indications: zero is low and 9 is high.	Confidence here is 8, a relatively high value.

*hr - hours: 00 to 24
mm - minutes: 00 to 59
ss - seconds: 00 to 59

TABLE 4 (1:71)

THE ADS EVENT MESSAGE STRING FOR PASS 2

EXAMPLE: ADS Pass 2 Message Set: //CWC/ADS//N2/142913/U/240/67/143318/9//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
1. Detection Source and Pass Number	N or S 2	North ADS Pass 2 Message
2. Time of Detection	hrmmss 000001 to 240000	North ADS sensor detection occurred at 1400 hours, 29 minutes and 13 seconds. ZULU.
3. Launcher Type	1 - Launcher Type 1 2 - Launcher Type 2 I - Indistinct U - Unknown Indistinct means the sensor intensity matches either Type 1 or Type 2 launchers. Unknown means the sensor intensity was too low or too high for Type 1 or Type 2 launchers.	Launcher type is unknown.
4. Heading	000-359 0° (North) to 359° East is 90°; South is 180°; West is 270°; XXX means no directional movement is discernable yet.	240° means the object is heading toward Us.
5. BSS Sites that Should or Could Detect Object	0 to 7 (two or more digits imply several sites). Zero is used if there is no movement or movement is away from Us.	Sites 6 and/or 7 should detect the object.

TABLE 4 (Cont'd)

THE ADS EVENT MESSAGE STRING FOR PASS 2

EXAMPLE: ADS Pass 2 Message Set: //CWC/ADS//N2/142913/U/240/67/143318/9//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
6. Expected Time the BSS Message Should Be Generated.	hrmmss 000001 to 240000 or XXXXXX If field is zero, this should be XXXXXX.	Object should be detected at the indicated time: 1400 hours, 33 minutes and 18 seconds, ZULU.
7. Sensor Confidence	0 to 9 The sensor site's confidence that the data <u>are</u> valid.	Confidence is 9, very high.

TABLE 5 (1:74)

THE BSS EVENT MESSAGE STRING FOR PASS 2

EXAMPLE: BSS Message Set: //CWC/BSS//7/143317/1/233/Y/143538/9//

DATA FIELD (In order, left to right)	CODES USED AND THEIR MEANING	INTERPRETATION OF THIS MESSAGE PROPERLY DECODED
1. Detection Source (BSS Site Number)	1 to 7	Barrier Surveillance Site Number 7 detected an object.
2. Detection Time	hrmmss 000001 to 240000	The BSS-7 detection occurred at 1400 hours, 33 minutes and 17 seconds ZULU.
3. Number of Objects	0 to 5	Only <u>one</u> object was detected by BSS-7.
4. Heading	000 to 359 0° (North) to 359° East is 90° South is 180° West is 270°	Object has a heading of 233°, roughly Southwest but toward Us.
5. Trajectory Stability	N - NO, maneuvering Y - YES, stable	A stable ballistic trajectory has been followed.
6. Expected Time of Impact	hrmmss 000001 to 240000	Object impact is anticipated at 1400 hours, 35 minutes, and 38 seconds ZULU.
7. Sensor Confidence	0 to 9 The BSS-7 site confidence that he detection data are valid indications; zero is low and 9 is high.	Site confidence is 9, very high.

Table 6

Listing of All the SENTINEL Actors

PROPERTY TAG	NAME	LATITUDE (in degrees)	LONGITUDE (in degrees)	TYPE
High Value Target List				
TGT1	LIVINGSTON	39	32	CAPITAL
TGT2	BRF	35	28	C3
TGT3	CDC	39	35.5	C3
TGT4	CWC	31	30	C3
TGT5	ADSGSF	33	35	C3
TGT6	BSSGSF	25.5	29	C3
Target Area 1				
TGT7	KEYNORTH	23.5	37.5	CITY
TGT8	RIVERTON	20.5	34.5	CITY
TGT9	NORTHCITY	20.5	30.5	CITY
TGT10	CLEAR	21	37.5	MILITARY
TGT11	MINOT	18	33.5	MILITARY
TGT12	CARLSBAD	22.5	38.5	INDUSTRIAL
TGT13	ARTOIS	21.5	30	INDUSTRIAL
TGT14	MANCHESTER	23	31	INDUSTRIAL
Target Area 2				
TGT15	TOLLCITY	25.5	33	CITY
TGT16	PINEGROVE	26	31	CITY
TGT17	OCEANCITY	28	39	CITY
TGT18	WARREN	26.5	36	MILITARY
TGT19	SEYMOUR	28	29	MILITARY
TGT20	MARATHON	26	36	INDUSTRIAL
TGT21	SAAR	27.5	35	INDUSTRIAL
TGT22	FRANCONIA	28.5	34	INDUSTRIAL
Target Area 3				
TGT23	HAYES	31.5	35.5	CITY
TGT24	WHITESANDS	32.5	37.5	CITY
TGT25	DELTA	29.5	30	CITY
TGT26	GEORGE	30.5	29	MILITARY
TGT27	LIDO	29.5	32.5	INDUSTRIAL

Table 6 (Cont'd)

Listing of All the SENTINEL Actors

PROPERTY TAG	NAME	LATITUDE	LONGITUDE	TYPE
Target Area 4				
TGT28	SOUTHRICH	36	34	CITY
TGT29	NORDEN	35.5	32	INDUSTRIAL
TGT30	EMDEN	36	33.5	INDUSTRIAL
TGT31	DAUN	37	32.5	INDUSTRIAL
TGT32	MUHL	39.5	34	INDUSTRIAL
TGT33	LOHR	40.5	35.5	INDUSTRIAL
BSS Sites				
TGT01	BSS#1	20	24.5	
TGT02	BSS#2	22	24.5	
TGT03	BSS#3	25	22.5	
TGT04	BSS#4	27.5	25.5	
TGT05	BSS#5	30	25	
TGT06	BSS#6	32.5	25	
TGT07	BSS#7	36	23.5	
Launch Sites				
LS1	KRASNOYARSK	21.5	20.5	GREEN
LS2	KAPUSTIN	23	21	GREEN
LS3	KURSK	22	13.5	RED
LS4	LODZ	25.5	19	GREEN
LS5	LIVONIA	28.5	14.5	RED2
LS6	LATVIA	29.5	13.5	RED1
LS7	LOYANG	29	12.5	RED1
LS8	MINSK	30.5	19	RED
LS9	MEMEL	31	11	RED
LS10	MARENGO	31.5	10.5	RED
LS11	MOSUL	34	20.5	GREEN
LS12	MUKDEN	34.5	17	RED
LS13	NARA	35.5	20.5	GREEN
LS14	NORI	38	18	RED2

Appendix C

Evaluation and Use of LOOPS

This discussion is based on material drawn from the LOOPS manual, reference 3. The examples used were developed by the author. LOOPS is an expert system building tool developed by the Xerox Corporation to help build knowledge based systems. LOOPS provides four different paradigms of programming, based on the premise that different paradigms are appropriate for different applications:

1). Procedure-oriented programming: This feature permits large procedures to use subroutines, by using the InterLISP-D facility on which LOOPS is built.

2). Object-oriented programming: Large objects are built up from small ones which combine instructions and data and send messages to each other. Inheritance is also permitted for objects by use of a class lattice.

3). Access-oriented programming: Uses a structure called an "active value" which can trigger additional computations when data is changed or read. They can be thought of as "probes" which can be placed on an object.

4). Rule-oriented programming: LOOPS organizes rules into rule sets which specify the rules and a control structure, which offers an "audit trail" facility which can trace all the invoked rules during processing (3:4).

During the course of this project, the author used mainly the rule oriented programming paradigm along with limited use of the

object oriented proramming paradigm. Due to the manner in which the SIMCOPE simulation was modeled, the SCENARIO REPRESENTATION was incompatible with the largely object oriented representation of LOOPS. In order to use the LOOPS rule base facilities it is necessary to first define objects and classes.

MetaClasses, Classes and Objects

The natural starting point for a beginner in using LOOPS is learning to define classes. It is difficult to use LOOPS without defining objects, but one cannot create an object which in the LOOPS terminology can be defined as "the specific instance of a class" without defining a class. One starts off by using a command to define a class. The AFITStudent class is defined by:

```
(DC 'AFITStudent)
```

Having defined the class, it can now be edited to permit "instance variables" to be attached to the class. Values can be attached to a class only if an instance variable has been defined for that class. The command to edit classes is: (EC 'AFITStudent)

After using the editor facility which uses the "mouse" as a pointer, the class AFITStudent is edited to provide instance variables for the values of graduation date, totalhours and address. Now, the output looks as follows:

```
[DEFCLASS AFITStudent
   (MetaClass Class Edited:     (* "24 August 1985 14:45"))
   (Supers Student Officers)
   (InstanceVariables
        (GradDate Dec85 doc     (* graduation date))
        (TotalHours 72 doc      (* quarter hours for graduation))
        (Address Dayton,Ohio doc    (* where AFIT students live))))
```

Comments in INTERLISP-D are preceeded by an asterisk and enclosed with parantheses. Starting with the second line, one notes that the class AFITStudent was edited at the stated time and date. The next line demonstrates inheritance; in this case, there are two other classes entitled STUDENT and OFFICERS that AFITStudent can "inherit" values from. Finally the last line shows the creation of three instance variables along with their default values. Therefore, the default value of Graduation Date for all AFITStudents is Dec85. The "doc" following the date is a "placeholder" used because the number of items in the Instance Variable lines must be an odd number, and in this case use of the "doc" brings all the instance variables to three items.

Having defined and edited the class AFITStudent, it is now possible to actually define an object. Objects are defined after classes are because the object is a specific instance of a class. The command to define DanTobat as an object of the class AFITStudent is:

```
(SETQ DanTobat (←($ AFITStudent) New))
```

This creates DanTobat to be a member of the class of AFITStudent. A command that checks whether an item is indeed a LOOPS object is:

```
(Object? DanTobat)
```

This will now be true (T), having created DanTobat to be a member of AFITStudent. Prior to the command that set DanTobat to be an object, this would have resulted in a NIL value. One of the most useful ways in using objects is being able to place and retrieve values from them. The command that retrieves values is:

```
(GetValue DanTobat 'Address)  results in: "Dayton,Ohio"
```

The second item in this command must be a LOOPS Object and the last item which is quoted is an Instance Variable name which has already been attached to the class of which the object is a member. This command results in "Dayton,Ohio" since this was the default value given to the AFITStudent class for the address instance variable. The GetValue and PutValue LOOPS commands are very similar to the GETPROP and PUTPROP commands of property lists in LISP. The way to attach a value to the object, rather than the use class default value is with PutValue:

```
(PutValue DanTobat 'GradDate  'Nov85)
```

This now changes the association of GradDate with the object to Nov85 instead of Dec85. However, the default value of the class remains unchanged. Thus, if BobRussel is also a member of the AFITStudent class:

(GetValue BobRussel 'GradDate)   results in: "Dec85"

Whereas (GetValue DanTobat 'GradDate) now results in: "Nov85"

Having discussed the concept of objects and classes, it is straight forward to extend the idea to the one of a metaclass. A Metaclass is a class whose instance variables are shared by several subclasses. Metaclasses were not used in this project. Objects were only used as the blackboards in SENTINEL, mainly because of their inability to use variable names for objects, unlike property lists which do permit this capability. As an example, suppose the LISP literal atom or variable TEST is set to BobRussel. Any time TEST is evaluated it results in BobRussel. Thus if this command is used:

```
(PUTPROP TEST 'NAME 'BOB)
```

The result is a property list under BobRussel whose value for the variable NAME will be BOB. Thus, these two commands in LISP are equivalent: (GETPROP TEST 'NAME)

or (GETPROP 'BobRussel 'NAME) result in:"Bob"

However this will not work with LOOPS commands, since the object where the value is placed on cannot be a variable. Thus,

(GetValue TEST 'GradDate) results in: "NIL"

LOOPS knows of no object named TEST. However,

(GetValue BobRussel 'GradDate) results in:"Dec85"

The inability to use variables in place of objects was a significant handicap to the concept of locating a specific launch site among many suspected launchsites, as well as having a variable number of launch sites or launch events. More elaborate ways of getting around this obstacle were tried, but they were defeated because LOOPS maps its objects and any values which may be evaluated to be objects to a specific memory location. Even if TEST is evaluated using more complex techniques, it results in a pointer to the memory location for the LOOPS object BobRussel. If TEST is now reset to DanTobat, all references to TEST for LOOPS still point to BobRussel even though the value for TEST has been changed. For these reasons, objects were not used widely in this project, but the rule based facitlities of LOOPS were used extensively.

Rule Sets

Before writing rule sets, a class must be defined that will serve as the "Workspace Class" for the rule set. If no classes have

been defined or an incorrect workspace class is given to the rule set, the rule set will not run. Please note that an object is an instantiation of a class, and not a class itself. Therefore, BobRussel cannot be given to a rule set as a workspace class, but AFITStudent can be. Rule sets are created by using:

(← $RuleSet New)

The user is then prompted for the rule set name which for this example will be "Regions." Having defined a new rule set, the rules are actually placed in it by editing the rule set. The command to do this is:

(← Regions ER)

Regions is the name of the rule set, and this sends the user to a menu based rule editor, whose menu can be invoked only by typing a (control) X or a (control) Z. If the editor does not invoke the menu upon use of a (control) X, this normally indicates that there is a LISP related problem in one of the rules such as a missing parantheses. In any case, (control) Z will always invoke the menu. This menu permits the user to compile the rule set into LISP, save it, return to the previous editor or actually view the translated LISP code of the rule set. At the top of the rule set is the control structure which after editing looks like this:

```
[Editing RULESET Regions
    Workspace Class: Launchsite;
    Compiler Options:  TT;
    Temporary Variables:  ;
    Control Structure: Dol;
    ***********
```

The first item which can be edited is the workspace class which must

be a previously defined LOOPS class. There are four compiler options which are useful for debugging purposes:

T: Trace if the rule satisfied.

TT: Trace if rule is tested.

B: Break if rule is satisfied.

BT: Break if rule is tested.

When this particular rule set is run the TT compiler option will show the user every rule that is tested. The temporary variables are used to define local variables that exist only within that rule set and are identical to the temporary variables used in a LISP PROG statement. LOOPS permits the rule set to have a varying control structure to control the execution of rules:

Do1: Exits when first rule satisfied is executed.

DoAll: All rules that are satisfied are executed.

While1: Cyclic version of Do1. Execute while one rule satisfied.

WhileAll: Cyclic version of DoAll.

This varying control structure permits the more efficient use of rulesets. Note that in order to use the While1 or WhileAll control structure the user must specify what the Iteration Condition is. Normally, T for True is used for this condition. The Iteration Condition is not located in the normal Rule Set editor control structure as was shown for the "Regions" rule set above. Instead, the user must select the "EditAllDecls" option from the rule base editor menu to edit this item.

The actual rules themselves are IF THEN constructs which permit

the use of regular LISP instructions as well as shorthand operations on LOOPS objects. The rule is ended by a semicolon. The most frequently used operations between variables or objects are the mathematical ones such as =, > and < or >= and <=. The equal operator can also be used with variables such as: "IF TEST = 'BobRussel." Other operators are available and are listed on pages 83 and 84 of the LOOPS manual. The following example shows the top rule in standard LOOPS LISP code and the bottom one utilizes shorthand notation for operations involving objects. The two rules are identical and accomplish the same thing:

```
  IF (GetValue BobRussel 'GradDate) = 'Dec85
THEN (PutValue BobRussel 'Hours 67);

  IF BobRussel:GradDate = 'Dec85
THEN BobRussel:Hours  67;
```

Thus, use of LOOPS objects does make the accessing and placing of values much easier. However, the same name cannot be used for both a LISP variable and an Instance Variable. In the example above, HOURS and GRADDATE <u>cannot</u> be used as LISP variables since they are instance variables of the BobRussel object. Note that each THEN statement is ended by a semicolon. Both the IF and THEN parts may have multiple arguments, which are interpreted as logical ANDs. For example, the rule: "IF A B C THEN D E;" is interpreted as:

IF A AND B AND C THEN D AND E.

A rule can also just have a THEN consequent with no IF part. In these cases, the rule is <u>always</u> executed. Execution of the rule set can always be terminated by a "(STOP)" statement, which overrides the control structure. Thus, use of a "THEN (STOP);" will cause any

execution which has reached that point to be stopped. The following figure illustrates a simple five rule set called Regions. Regions maps the reported launchsite into the country of THEM using only five rules. Notice that the last statement is a THEN not preceeded by an IF statement. That coupled with the Dol control structure specified at the top of the rule set indicates that if no rule has been triggered previously, it is assumed that the launchsite must belong to the central region. The page afterward demonstrates the use of the TT compile option which results in a printout of every rule as it is tested and finally the execution of the last rule.

To actually run a rule set, use the command:

(RunRS rulesetname workspace)

The workspace is a LOOPS object that the rule set acts upon. Since Regions does not act upon any rule set (instead the LISP variable REGION is set to some value), it is sufficient to use:

(RunRS 'Regions)

The value of the evaluated rule set is the value of the final rule executed. If no rule is executed the value is NIL. The RunRS command can also be used as part of an IF or THEN statement. Rulesets which control the triggering of other rule sets are termed metarule sets. One final command to use in rulesets is to print them out using:

(← rulesetname PPRules)

There are several pitfalls in using rule sets. For one, in the "Buttress" release of LOOPS there is no scroll bar for editing rule sets. As a result one loses the ability to edit the control structure or the first few rules unless rule sets are kept very

```
9←(← Regions PPRules)
Using $Regions
RuleSet Regions
Last Edited: 28-Oct-85 14:25:08 by
WorkSpace Class: Launchsite;
Compiler Options:  ;
Temporary Vars:  ;
Control Structure: DO1;
********



(* This rule set determines what region in the country of T
HEM the launchsite is located in.)

IF LNCHLAT <= 26
   LNCHLONG <= 17
THEN (SETQ REGION 'OUTT);

IF LNCHLAT <= 26
   LNCHLONG > 17
THEN (SETQ REGION 'INN);

IF LNCHLAT > 26
   LNCHLONG >= 18
   LNCHLAT < 33
THEN (SETQ REGION 'WEST);

IF LNCHLAT >= 33
   LNCHLONG > 19
THEN (SETQ REGION 'WEST);

IF LNCHLAT >= 33
   LNCHLONG <= 19
THEN (SETQ REGION 'SOUTH);

(* If launchsite does not fall in the above regions, it is
assumed to be located in the CENTRAL region)

THEN (SETQ REGION 'CENTRAL);
#&(RuleSetSource "LZU0.zI[:.AK9.4")
```

Figure 29. Printout of the Regions Rule Set.

```
43←(RunRS 'Regions)
Testing rule 1 in  RuleSet Regions

(* This rule set determines what region in the country of THEM th
e launchsite is located in.)

IF LNCHLAT <= 26
  LNCHLONG <= 17
THEN (SETQ REGION 'OUTT);

    ... Rule 1 from Regions
        edited by  on  6-Nov-85 13:14:51

Testing rule 2 in  RuleSet Regions

IF LNCHLAT <= 26
  LNCHLONG > 17
THEN (SETQ REGION 'INN);

    ... Rule 2 from Regions
        edited by  on  6-Nov-85 13:14:51

Testing rule 3 in  RuleSet Regions

IF LNCHLAT > 26
  LNCHLONG >= 18
  LNCHLAT < 33
THEN (SETQ REGION 'WEST);

    ... Rule 3 from Regions
        edited by  on  6-Nov-85 13:14:51

Testing rule 4 in  RuleSet Regions

IF LNCHLAT >= 33
  LNCHLONG > 19
THEN (SETQ REGION 'WEST);

    ... Rule 4 from Regions
        edited by  on  6-Nov-85 13:14:51

Executing rule 4 in RuleSet Regions

IF LNCHLAT >= 33
  LNCHLONG > 19
THEN (SETQ REGION 'WEST);

    ... Rule 4 from Regions
        edited by  on  6-Nov-85 13:14:51

WEST
```

Figure 30. Printout of the Testing and Execution of the Regions Rule Set.

small or the bottom part of the rule set is deleted and then rewritten after the changes are made. The other shortcoming is the lack of an logical OR capability. Instead of being able to use:

IF A AND (B OR C) THEN D;

one must write two separate rules:

IF A AND B THEN D;   IF A AND C THEN D;

This is also true for values that fall into discontinous ranges. For example in searching for a latitude that lies below 23 degrees and above 30 degrees, one is tempted to use:

```
  IF LATITUDE <= 23
     LATITUDE > 30
THEN (SETQ REGION 'POLAR);
```

But this rule will never be triggered since it would require a latitude that is both less than 23 degrees yet higher than 30 degrees. One must write two rules to cover this case as well. Care must be taken when using mathematical operators that the variable used is always a number. For example, if LATITUDE is not set, its value will be NIL, and using NIL with > and < will result in an error. The solution is to check and if LATITUDE is NIL then set LATITUDE to 0 or some number that will not trigger a rule improperly.

In spite of these shortcomings, the LOOPS rule based facility was used extensively during this project because it does speed up programming time considerably. The LOOPS manual itself is difficult for the beginner to use because it is not laid out in the order that one would use it and because its examples cannot easily be

duplicated, and are inconsistent from chapter to chapter. There are other facilities available in the LOOPS package such as active values, and knowledge bases. However those facilities were of limited value to this project. The knowledge base facilities for example, stcre the values of diverse LOOPS objects and their variables at different points in time.

To summarize, LOOPS is a very good tool for object oriented programming that has some useful capabilities in other areas such as rule bases. The major limitations of the "Buttress" release are the lack of a sroll bar when editing rule sets, and the inability to easily use variables with objects. These factors made the SENTINEL rule sets smaller and more numerous than they would have been otherwise, and forced SENTINEL to rely on property lists for its actors instead of converting the actors to objects. In spite of its flaws, only thorugh the use of the LOOPS expert systems building tool was it possible to complete SENTINEL in the time alloted for this study.

Appendix D

Miscellaneous Functions and Program Listing

The intent of this section is to cover some of the SENTINEL functions and rule sets, which although valuable, were not discussed in the main text. At the end of this section is located the listing of every LISP program and LOOPS rule set used in SENTINEL.

ADDTARGET

ADDTARGET is a function which makes it easier for the MWO to add targets to the SENTINEL scenario. It asks the user a series of questions such as "What is the latitude of the Target?" to which the user responds. A new target actor is thus created and a property list containing all the values that the MWO replied to. This target actor is automatically included in all the follow on scenario processing since it is added to FANCALL for every launch site actor. The RANGE and HEADING to this target are therefore computed for any launch events which occur after ADDTARGET has been invoked.

COMBINATORIAL

COMBINATORIAL is a LISP function which is used to combine the evidence of two factors into a single combined factor. The first factor is the "strong" factor which more greatly affects the outcome of the result than the second "weak" factor. Each factor is expressed as a number with a value ranging from 1 to 4 where 4 is the highest state possible and 1 is the lowest. The following

demonstrates the results of COMBINATORIAL, using the symbol @ to represent the process, and where the first number stated is the "strong" factor:

4 @ 4 = 4	3 @ 4 = 3	2 @ 4 = 3	1 @ 4 = 2
4 @ 3 = 4	3 @ 3 = 3	2 @ 3 = 2	1 @ 3 = 2
4 @ 2 = 3	3 @ 2 = 2	2 @ 2 = 2	1 @ 2 = 1
4 @ 1 = 3	3 @ 1 = 2	2 @ 1 = 1	1 @ 1 = 1

Note that COMBINATORIAL is not a way of reasoning with uncertainty. Instead, it is a useful method of combining facts easily and quickly in a numerical fashion to produce a higher level of abstraction. The same results could have also been found by simply enumerating all possible states of occurences. The result has "noise" in it but since all these facts and results are discrete so there is no probability involved. A state is simply achieved or not achieved. As an example of using COMBINATORIAL, consider attempting to arrive at a more abstract general state by combining two facts: the number of sensors that are OUT with ZERO capability with those that are AFFECTED with PARTIAL capability. In this case, the number of sensors that are OUT is obviously of more concern than the number of AFFECTED sensors so that OUT is the "strong" factor. Assuming the state of OUT is at a 4 and the state of affected is at a 2, which result in an overall 3. Before the combination, two statements could be made:

Number of OUT sensors is HIGH, there is cause for ALARM.

Some AFFECTED sensors are reported, CAUTION is recommended.

Following COMBINATORIAL a more abstract, general statement can be made which although not as accurate as the above two summarized the

situation succintly:

WARNING: Overall sensor status DEGRADED.

COMBINATORIAL was used mainly to produce the abstract states which were interpreted as comments.

CONVERT

CONVERT is a LISP function which makes it easier for the MWO to understand the output of SENTINEL. Since SENTINEL launch site and target actors are not identified by their name, the original output features the use of the property tags of the actors such as TGT6, LS8 and so on. CONVERT takes a list or a single value in the property tag format, and converts it to its NAME property on the property list. Thus, the list "(TGT7 TGT8 TGT9)" is converted to "(KEYNORTH RIVERTON NORTHCITY)."

HELP

HELP is the on line operator's manual to SENTINEL which is intended to assist the user in SENTINEL operations. It is invoked by typing in "(HELP)" and gives the user a short introduction to the major features provided in SENTINEL. A short troubleshooting section is included for the use as a guide to several items which may malfunction when first installing SENTINEL on a LISP machine.

MODIFY

MODIFY is a LISP function which is of great value in the training or troubleshooting mode. MODIFY takes the SIMCOPE message which has

been most recently processed and permits the user to MODIFY almost any parameter of an ADS-1, ADS-2 or BSS message. For an ADS-1 message, the sensor infrared intensity, site confidence, reported launch site coordinates and the status of the reported sensor can all be modified. Invoked when the user types in (MODIFY), the program informs the user of what parameter is currently under consideration, what its previous value is and what the acceptable range of values of the parameter are. The user can type in the new value which is automatically incorporated into the MESSAGE. Typing in NIL after the first entry results in SENTINEL processing the modified message. NIL on the first entry exits from the program without invoking SENTINEL again.

Since the message is actually modified by use of the RPLACA and the RPLACD commands, the old value for the message is destroyed. MODIFY does this function for ADS-1 messages while MODIFY2 and MODIFY3 perform the function for ADS-2 and BSS messages respectively. INT and SYS messages cannot be modified. It is important to note that only the most recently processed message can be modified automatically. If an ADS-1 EVENT3 message is followed by an SYS message, typing in (MODIFY) will result in a statement that "SYS messages cannot be modified." In order to use MODIFY in those circumstances, MESSAGE must be set to the message which caused EVENT3, and EVENTMATCH must be set to EVENT3 if it is not EVENT3 already.

SENTINEL

SENTINEL is the initialization LISP function. Typing in (SENTINEL) results in all blackboards, launch sites, and target actors being returned to their original state. Counters are also initialized and if the HIVALTAG of any launch site actor is blank, SENTINEL will process the HIVALTAG using HIVALCALL and embed the result on the appropriate launch site actor. EVENTs and the EVENTLIST and MESSAGE are all set to NIL and the statement: "SENTINEL DECISION AID INITIALIZED" appears to inform the MWO when initialization is complete.

Program Listing

What follows is the complete listing of all the SENTINEL functions and rule sets. All together, SENTINEL has over 800 rules. The index to this listing is located on page viii, near the beginning of this report.

```
(ADDTARGET
  [LAMBDA NIL                                    (* edited: "11-Oct-85 14:24")

               (* ADDTARGET: permits the addition or updating of a target by readi
ng in responses. Updates the property list of the launchiste, and creates a new pro
pety list if required. Also updates the target area list inputs: No inputs required. To
use type (ADDTARGET OUTPUTS: a property list under the tag of TGT NUMBER c
onsisting of (TGTNUM NAME XXXX LAT XX LONG XX TYPE XXXX)))


    (PROG (READNAME READNUM READLAT READLONG READTYPE)
          (printout T "LATEST TARGET TO BE ADDED WAS "
                    LASTGT T)
          (printout T
         "Type NIL if you don't wish to add a target."
                    T)
          (printout T
         "What is TARGET number ? (Begin with a TGT) "
                    T)
          (SETQ READNUM (RATOM NIL NIL))
          (COND
            ((NULL READNUM)
              (RETURN)))
          (SETQ LASTGT READNUM)
          (printout T "WHAT IS TARGET NAME ?" T)
          (SETQ READNAME (RATOM NIL NIL))
          (PUTPROP READNUM (QUOTE NAME)
                   READNAME)
          (printout T "What is target LATITUDE ?" T)
          (SETQ READLAT (RATOM NIL NIL))

               (* Decide what target area it falls into by these rules: Target Latitud
e < 24 degrees = Area 1 Target Latitude > 34 degrees = Area 4 Target Latitude >
 29 but <34 = Area 3 Target Latitude < 29 but >24 = Area 2)


          (PUTPROP READNUM (QUOTE LAT)
                   READLAT)
          (printout T "What is target LONGITUDE ? " T)
          (SETQ READLONG (RATOM NIL NIL))
          (PUTPROP READNUM (QUOTE LONG)
                   READLONG)
          (printout T
"What is TARGET type ? (City, Military, Industrial or C3)
"
                    T)
          (SETQ READTYPE (RATOM NIL NIL))
          (PUTPROP READNUM (QUOTE TYPE)
                   READTYPE)
          (printout T "PROPERTY LIST UNDER " READNUM
                " CREATED, WITH THE FOLLOWING VALUES."
                   T)
         (RETURN (GETPROPLIST LASTGT])
```

using $ADS1Confidence

RuleSet ADS 1Confidence

Last Edited; 3-Oct-85 17:58:42 by

```
WorkSpace Class: Sensors;
Compiler Options:  ;
Temporary Vars:   P1SUM P1LSTYPE;
Control Structure: DOALL;
********



(* The following RuleSet uses 1IRINTENSITY, 1SENSORCONF, RE
SULT and SENSORNUMBER to compute an ADS1 confidence)

IF P1IRINTENSITY < 2
THEN (PUTPROP EVENTCOUNT 'P1IRTYPE 'UNKNOWN)
     (SETQ P1SUM 0);

IF P1IRINTENSITY > 7
THEN (PUTPROP EVENTCOUNT 'P1IRTYPE 'UNKNOWN)
     (SETQ P1SUM 0);

IF P1IRINTENSITY >= 2
   P1IRINTENSITY < 5
THEN (PUTPROP EVENTCOUNT 'P1IRTYPE 'TYPE1)
     (SETQ P1SUM 2)
     (SETQ P1LSTYPE (GETPROP RESULT 'TYPE));

IF P1IRINTENSITY >= 5
   P1IRINTENSITY < 6
THEN (PUTPROP EVENTCOUNT 'P1IRTYPE 'INDISTINCT)
     (SETQ P1SUM 1);

IF P1IRINTENSITY >= 6
   P1IRINTENSITY <= 7
THEN (PUTPROP EVENTCOUNT 'P1IRTYPE 'TYPE2)
     (SETQ P1SUM '2)
     (SETQ P1LSTYPE (GETPROP RESULT 'TYPE));

(* This rule matches the IR intensity reading to the type o
f missiles reported at the launchsite.)

IF (GLC P1IRTYPE) = (GLC P1LSTYPE)
THEN (SETQ P1SUM (PLUS P1SUM 1));

(* The following classify reported sensor confidence into c
ategories of HIGH, MEDUIM or LOW.)

IF P1SENSORCONF >= 7
THEN (SETQ P1SUM (PLUS P1SUM 3))
```

```
IF P1SENSORCONF >= 7
THEN (SETQ P1SUM (PLUS P1SUM 3))
     (PUTPROP EVENTCOUNT 'P1SENSORCONF 'HIGH);

IF P1SENSORCONF < 7
   P1SENSORCONF >= 4
THEN (SETQ P1SUM (PLUS P1SUM 2))
     (PUTPROP EVENTCOUNT 'P1SENSORCONF 'MEDUIM);

IF P1SENSORCONF < 4
THEN (PUTPROP EVENTCOUNT 'P1SENSORCONF 'LOW)
     (SETQ P1SUM (PLUS P1SUM 1));

(* The following modify the confidence level according to w
hat the CAPABILITY of the reported satellite is: FULL PARTI
AL or LOW.)

IF SENSORNUMBER = 'S1
THEN (SETQ P1SENSORCAP (GETPROP 'ADSSOUTH 'CAPABILITY))
     (PUTPROP EVENTCOUNT 'P1SENSORTYPE 'ADSSOUTH);

IF SENSORNUMBER = 'N1
THEN (SETQ P1SENSORCAP (GETPROP 'ADSNORTH 'CAPABILITY))
     (PUTPROP EVENTCOUNT 'P1SENSORTYPE 'ADSNORTH);

IF P1SENSORCAP = 'NIL
THEN (PUTPROP EVENTCOUNT 'P1SENSORCAP 'FULL);

IF P1SENSORCAP = 'FULL
THEN (PUTPROP EVENTCOUNT 'P1SENSORCAP 'FULL);

IF P1SENSORCAP = 'PARTIAL
THEN (SETQ P1SUM (SUB1 P1SUM))
     (PUTPROP EVENTCOUNT 'P1SENSORCAP 'PARTIAL);

IF P1SENSORCAP = 'ZERO
THEN (SETQ P1SUM (PLUS P1SUM -2))
     (PUTPROP EVENTCOUNT 'P1SENSORCAP 'ZERO);

(* The following permits INTELBoard to change the confidenc
e level.)

THEN (SETQ P1INTSOURCE (GetValue IntelBoard 'SOURCE))
 (SETQ P1INTCOND (GetValue IntelBoard 'INTELCONDITION));

IF P1INTCOND = 'RED
   P1INTSOURCE = 'OPEN
THEN (SETQ P1SUM (PLUS P1SUM 1));

IF P1INTCOND = 'RED
```

```
IF P1INTCOND = 'RED
   P1INTSOURCE = 'INTEL
THEN (SETQ P1SUM (PLUS P1SUM 2));

IF P1INTCOND = 'YELLOW
   P1INTSOURCE = 'INTEL
THEN (SETQ P1SUM (PLUS P1SUM 1));

IF P1INTCOND = 'GREEN
THEN (SETQ P1SUM (SUB1 P1SUM));

IF P1INTCOND = 'TEST
   P1INTSOURCE = 'INTEL
THEN (SETQ P1SUM (PLUS P1SUM 1));

(* The following permits the status of the launchsite, KNOW
N, UNKOWN or SUSPECTED to influence the confidence.)

THEN (SETQ P1LSTYPE (GETPROP RESULT 'TYPE));

IF P1LSTYPE = 'GREEN
THEN (PUTPROP EVENTCOUNT 'P1LSTYPE 'SUSPECTED);

IF P1LSTYPE = 'NIL
THEN (PUTPROP EVENTCOUNT 'P1LSTYPE 'UNKNOWN);

IF P1LSTYPE ~= 'NIL
   P1LSTYPE ~= 'GREEN
THEN (PUTPROP EVENTCOUNT 'P1LSTYPE 'KNOWN);

(* Finally, P1SUM is assigned to categories of confidence.)

IF P1SUM >= 6
THEN (PUTPROP EVENTCOUNT 'P1CONFIDENCE 'EXTHIGH);

IF P1SUM < 6
   P1SUM >= 4
THEN (PUTPROP EVENTCOUNT 'P1CONFIDENCE 'HIGH);

IF P1SUM = 3
THEN (PUTPROP EVENTCOUNT 'P1CONFIDENCE 'MIXED);

IF P1SUM = 2
THEN (PUTPROP EVENTCOUNT 'P1CONFIDENCE 'LOW);

IF P1SUM <= 1
THEN (PUTPROP EVENTCOUNT 'P1CONFIDENCE 'EXTLOW);
#&(RuleSetSource "LZU0.zI[:.AK9.3")
61←
```

```
using $ADS1FeatureSet
```

RuleSet ADS1FeatureSet

Last Edited: 3-Oct-85 18:00:57 by

```
WorkSpace Class: Targets;
Compiler Options:  ;
Temporary Vars:  ;
Control Structure: DOALL;
********


(* This rule set accepts the input of an event list and ret
urns the feature set of the event for the ADS1 event)

THEN (SETQ P1IRTYPE (GETPROP EVENTCOUNT 'P1IRTYPE))
 (SETQ P1SENSORCONF (GETPROP EVENTCOUNT 'P1SENSORCONF));

IF SENSORNUMBER = 'S1
THEN (SETQ P1SENSORCAP (GETPROP 'ADSSOUTH 'CAPABILITY));

IF SENSORNUMBER = 'N1
THEN (SETQ P1SENSORCAP (GETPROP 'ADSNORTH 'CAPABILITY));

IF P1SENSORCAP = 'FULL
   P1SENSORCONF = 'HIGH
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 9);

IF P1SENSORCAP = 'FULL
   P1SENSORCONF = 'MEDUIM
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 8);

IF P1SENSORCAP = 'PARTIAL
   P1SENSORCONF = 'HIGH
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 7);

IF P1SENSORCAP = 'FULL
   P1SENSORCONF = 'LOW
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 6);

IF P1SENSORCAP = 'PARTIAL
   P1SENSORCONF = 'MEDUIM
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 5);

IF P1SENSORCAP = 'ZERO
   P1SENSORCONF = 'HIGH
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 4);

IF P1SENSORCAP = 'PARTIAL
   P1SENSORCONF = 'LOW
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 3);
```

```
IF P1SENSORCAP = 'PARTIAL
   P1SENSORCONF = 'LOW
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 3);

IF P1SENSORCAP = 'ZERO
   P1SENSORCONF = 'MEDUIM
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 2);

IF P1SENSORCAP = 'ZERO
   P1SENSORCONF = 'LOW
THEN (PUTPROP EVENTCOUNT 'P1SENSOR 1);

(* Finally, a bit of housecleaning is done)
THEN (REMPROP EVENTCOUNT 'P1LSTYPE)
     (REMPROP EVENTCOUNT 'P1SENSORTYPE)
     (REMPROP EVENTCOUNT 'P1SENSORCAP)
     (REMPROP EVENTCOUNT 'P1SENSORCONF);
#&(RuleSetSource "GEU0.zI[:.W2<.3")
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←
NIL
68←(REDO 1)
```

using $ADS2Confidence

RuleSet ADS2Confidence

Last Edited: 3-Oct-85 18;18;37 by

```
WorkSpace Class: Launchsite;
Compiler Options:  ;
Temporary Vars:   P1CONFIDENCE;
Control Structure: DOALL;
********



(* ADS2Confidence interprets the ADS2 sensorconfidence, the
n decomposes ADS1 Confidence to a number (P2SUM) which then
 calls on ADS2FeatureSet, which returns the revised P2SUM.
This rules set interprets the new P2SUM to a P2CONFIDENCE,
shows output and calls on the confidence processing)


IF P2SENSORCONF >= 7
THEN (SETQ P2SCONF 'HIGH);

IF P2SENSORCONF >= 4
   P2SENSORCONF < 7
THEN (SETQ P2SCONF 'MEDUIM);

IF P2SENSORCONF < 4
THEN (SETQ P2SCONF 'LOW);

THEN (SETQ P1CONFIDENCE (GETPROP EVENTMATCH 'P1CONFIDENCE))
;

IF P1CONFIDENCE = 'EXTHIGH
THEN (SETQ P2SUM 5);

IF P1CONFIDENCE = 'HIGH
THEN (SETQ P2SUM 4);

IF P1CONFIDENCE = 'MIXED
THEN (SETQ P2SUM 3);

IF P1CONFIDENCE = 'LOW
THEN (SETQ P2SUM 2);

IF P1CONFIDENCE = 'EXTLOW
THEN (SETQ P2SUM 1);

THEN (RunRS 'ADS2FeatureSet);

IF P2SUM >= 5
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'EXTHIGH);
```

```
IF P2SUM = 4
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF P2SUM = 3
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'MIXED);

IF P2SUM = 2
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

IF P2SUM <= 1
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'EXTLOW);

THEN (RunRS 'TrendCheck)
     (PUTPROP EVENTMATCH 'P2HYPO NIL) (CF)
 (PRINTOUT T T "ADS2 REPORT ACKNOWLEDGED" T T
     "ADS2 EVENT has been matched with: " EVENTMATCH T T "S
ENTINEL DECISION AID CONFIDENCE is: " (GETPROP EVENTMATCH '
P2CONFIDENCE) T T "Further processing now underway...." T T
)
   EXPLAINBoard:COMMENT←'NIL  EXPLAINBoard:EXPFACT←'NIL;

(* The different CONFIDENCE levels are mapped to their
separate ways for processing. STOP placed after each     to
 avoid belief revision triggering more than one.)

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN (RunRS 'LOWConfidence)
     (STOP);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'MIXED
THEN (RunRS 'MIXEDConfidence)
     (STOP);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN (RunRS 'HIGHConfidence)
     (STOP);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN (RunRS 'EXTHIGHConfidence)
     (STOP);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
THEN (RunRS 'EXTLOWConfidence);
#&(RuleSetSource "GIU0.zI[:.;7:.3")
75←
NIL
75←
NIL
75←(REDO 100)
```

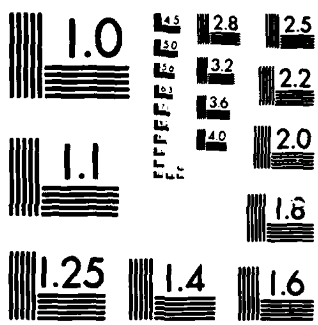
using $ADS2FeatureSet

RuleSet ADS2FeatureSet

```
Last Edited: 3-Oct-85 18:19:43 by
WorkSpace Class: Launchsite;
Compiler Options: ;
Temporary Vars:    P2DIFF P1IRTYPE P1SENSOR P2SENSORCAP P2SE
NSOR P2NEGDIFF;
Control Structure: DOALL;
********



(* This ruleset takes P1CONFIDENCE, P2 IR TYPE and SENSOR c
apability and reduces it to the ADS2FeatureSet which is P2C
ONFIDENCE, P2TYPE, and P2SENSOR)

THEN (SETQ P1IRTYPE (GETPROP EVENTMATCH 'P1IRTYPE))
     (SETQ P1SENSOR (GETPROP EVENTMATCH 'P1SENSOR));

(* sensornumber determines which satellite picked it up)
IF SENSORNUMBER = 'S2 THEN (SETQ P2SENSORCAP (GETPROP 'ADSS
OUTH 'CAPABILITY));

 IF SENSORNUMBER = 'N2 THEN (SETQ P2SENSORCAP (GETPROP 'ADS
NORTH 'CAPABILITY));

 (* P2SENSOR is determined here in the exact same manner P1
SENSOR was found) IF P2SENSORCAP = 'FULL P2SCONF = 'HIGH TH
EN (SETQ P2SENSOR 9);

 IF P2SENSORCAP = 'FULL P2SCONF = 'MEDUIM THEN (SETQ P2SENS
OR 8);

IF P2SENSORCAP = 'PARTIAL P2SCONF = 'HIGH
THEN (SETQ P2SENSOR 7);

IF P2SENSORCAP = 'FULL P2SCONF = 'LOW
THEN (SETQ P2SENSOR 6);

IF P2SENSORCAP = 'PARTIAL  P2SCONF = 'MEDUIM
THEN (SETQ P2SENSOR 5);

IF P2SENSORCAP = 'ZERO P2SCONF = 'HIGH
THEN (SETQ P2SENSOR 4);

IF P2SENSORCAP = 'PARTIAL P2SCONF = 'LOW
THEN (SETQ P2SENSOR 3);

IF P2SENSORCAP = 'ZERO P2SCONF = 'MEDUIM
THEN (SETQ P2SENSOR 2);
```

```
IF P2SENSORCAP = 'ZERO P2SCONF = 'LOW
THEN (SETQ P2SENSOR 1);

(* Difference in confidence levels between P1SENSOR and P2S
ENSOR checked. P2NEGDIFF set to number for use with ops)THE
N (SETQ P2DIFF (DIFFERENCE P2SENSOR P1SENSOR)) (SETQ P2NEGD
IFF 0) (PUTPROP EVENTMATCH 'P2SENSOR P2SENSOR);

 IF P2DIFF < 0 THEN (SETQ P2NEGDIFF (ABS P2DIFF));

 IF P2DIFF >= 1 P2DIFF <= 2
THEN (SETQ P2SUM (PLUS P2SUM 1));

 IF P2DIFF >= 3 THEN (SETQ P2SUM (PLUS P2SUM 2));

 IF P2NEGDIFF >= 1  P2NEGDIFF <= 2 THEN (SETQ P2SUM (SUB1 P
2SUM));

 IF P2NEGDIFF >= 3
THEN (SETQ P2SUM (PLUS P2SUM -2));

 (* The IR readings for pass 1 and pass 2 are compared, the
y fall into categories of MATCH, NOMATCH or MISMATCH) IF P1
IRTYPE = 'TYPE1 P2IRTYPE = 'U THEN (SETQ P2SUM (SUB1 P2SUM)
)  (PUTPROP EVENTMATCH 'P2TYPE 'MISMATCH);

IF P1IRTYPE = 'TYPE1 P2IRTYPE = 'I THEN (SETQ P2SUM (SUB1 P
2SUM)) (PUTPROP EVENTMATCH 'P2TYPE 'MISMATCH);

 IF P1IRTYPE = 'TYPE2 P2IRTYPE = 'U THEN (SETQ P2SUM (SUB1
P2SUM)) (PUTPROP EVENTMATCH 'P2TYPE 'MISMATCH);

IF P1IRTYPE = 'TYPE2 P2IRTYPE = 'I THEN (SETQ P2SUM (SUB1 P
2SUM)) (PUTPROP EVENTMATCH 'P2TYPE 'MISMATCH);

IF P1IRTYPE = 'INDISTINCT P2IRTYPE = '1 THEN (SETQ P2SUM (P
LUS P2SUM 1)) (PUTPROP EVENTMATCH 'P2TYPE 'NOMATCH);

IF P1IRTYPE = 'UNKNOWN P2IRTYPE = '2 THEN (SETQ P2SUM (PLUS
 P2SUM 1)) (PUTPROP EVENTMATCH 'P2TYPE 'NOMATCH);

 IF P1IRTYPE = 'INDISTINCT P2IRTYPE =  '2 THEN (SETQ P2SUM
(PLUS P2SUM 1)) (PUTPROP EVENTMATCH 'P2TYPE 'NOMATCH);

IF P1IRTYPE = 'UNKNOWN  P2IRTYPE = '1 THEN (SETQ P2SUM (PLU
S P2SUM 1)) (PUTPROP EVENTMATCH 'P2TYPE 'NOMATCH);

IF P1IRTYPE = 'TYPE1  P2IRTYPE = '2 THEN (PUTPROP EVENTMATC
H 'P2TYPE 'NOMATCH);
```

AD-A163 827 AN EXPERT SYSTEM DECISION AID FOR A COMMAND CONTROL AND 3/4
COMMUNICATIONS OPERATOR(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D L TOBAT
UNCLASSIFIED DEC 85 AFIT/GE/ENG/85D-50 F/G 17/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

```
IF P1IRTYPE = 'TYPE1  P2IRTYPE = '2 THEN (PUTPROP EVENTMATC
H 'P2TYPE 'NOMATCH);

IF P1IRTYPE = 'TYPE2 P2IRTYPE = '1 THEN (PUTPROP EVENTMATCH
 'P2TYPE 'NOMATCH);

THEN (SETQ P1LS (GETPROP EVENTMATCH 'P1LS));

(* Retrieve the launchsite above, see if new IRTYPE matches
 the reported TYpe of the launchsite) IF P2IRTYPE = (GLC (G
ETPROP P1LS 'TYPE)) THEN (SETQ P2SUM (PLUS P2SUM 1));

 IF (GETPROP EVENTMATCH 'P2TYPE) = 'NIL THEN (PUTPROP EVENT
MATCH 'P2TYPE 'MATCH);
#&(RuleSetSource "GIU0.zI[:.;7:.4")
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←
NIL
87←(REDO 100)
```

```
USE(PP ADS2HANDLER)
```

(ADS2HANDLER

```
  [LAMBDA (ADS2REPORT)                    (* edited: "18-Sep-85 13:36")
    (PROG (MATCHLIST MATCHPOINT MATCHTIME TIMEDIFF
                     OLDTIMEDIFF LATLNCH P1LS)

          (* This function first decomposes the ADS2 Report into its constituen
t parts and then attempts to match the record of all existing ADS1 Reports located
in EVENTLIST with the ADS2 Report)


          (SETQ MSG ADS2REPORT)
          (SETQ P2TIME (CAADR MSG))
          (SETQ P2SENSORCONF (CADDR MSG))
          (SETQ P2IRTYPE (CADAR (CDADAR MSG)))
          (SETQ P2HEADING (CADDR (CADAR MSG)))
          (SETQ P2BSS (CADDAR MSG))
          (printout T "This is P2TIME: " P2TIME T
                    "This is P2SENSORCONF: "
                    P2SENSORCONF T "This is P2HEADING: "
                    P2HEADING T "This is P2BSS: " P2BSS T
                    "This is P2IRTYPE: "
                    P2IRTYPE T)

          (* The matching part begins here; a time difference of more than 5
minutes (500) is held to be excessive in the missile warning scenario)


          (SETQ MATCHLIST EVENTLIST)
          (SETQ OLDTIMEDIFF 500)
          (SETQ EVENTMATCH NIL)
      LOOP(SETQ MATCHPOINT (CAR MATCHLIST))
          (COND
            ((NULL MATCHPOINT)
              (RETURN)))
          (SETQ MATCHTIME (GETPROP MATCHPOINT
                                   (QUOTE P1TIME)))
          [COND
            ((NULL MATCHTIME)
              (SETQ TIMEDIFF 501))
            (T (SETQ TIMEDIFF (ABS (DIFFERENCE PRESENTIME
                                               MATCHTIME]

          (* The closest match in terms of how close it is in time is stored as
OLDTIMEDIFF; if a better one comes along, EVENTMATCH is updated accordingly)


          (SETQ P1LS (GETPROP MATCHPOINT (QUOTE P1LS)))
          (SETQ LATLNCH (GETPROP P1LS (QUOTE LAT)))
```

(* The launch latitude reported by the satellite is checked to see if that latitude is normally covered by the satellite. If there is a problem matching the two, no match is made. Also, SENSORCAP and REASONOUT are set from the reporting satellite for inclusion on the SYSBoard.)

```
[COND
  ((LESSP TIMEDIFF OLDTIMEDIFF)
    [COND
      ((EQ (GNC SENSORNUMBER)
           (QUOTE S))
        (COND
          ((GREATERP LATLNCH 27.6)
            (SETQ OLDTIMEDIFF TIMEDIFF)
            (SETQ EVENTMATCH MATCHPOINT)
            (SETQ SENSORCAP (GETPROP
                 (QUOTE ADSSOUTH)
                 (QUOTE CAPABILITY)))
            (SETQ REASONOUT (GETPROP
                 (QUOTE ADSSOUTH)
                 (QUOTE REASON]
    (COND
      ((EQ (GNC SENSORNUMBER)
           (QUOTE N))
        (COND
          ((LESSP LATLNCH (QUOTE 29.6))
            (SETQ OLDTIMEDIFF TIMEDIFF)
            (SETQ EVENTMATCH MATCHPOINT)
            (SETQ SENSORCAP (GETPROP
                 (QUOTE ADSNORTH)
                 (QUOTE CAPABILITY)))
            (SETQ REASONOUT (GETPROP
                 (QUOTE ADSNORTH)
                 (QUOTE REASON]
(SETQ MATCHLIST (CDR MATCHLIST))
(GO LOOP))
```

(* If no match is found, EVENTMATCH will be NIL and situation can not progress any further because of lack of launch latitude and launch longitude information. SENTINEL announces this information to the user and lets him take over.)

```
(COND
  ((NULL EVENTMATCH)
    (CF)
    (printout T T T
           "NO ADS1 EVENT MATCHES THIS ADS2 REPORT"
               T "SPURIOUS ADS2 REPORT SUSPECTED" T
"SENTINEL DECISION AID CANNOT PROCESS THIS REPORT"
               T T
```

```
  (printout T T T
        "NO ADS1 EVENT MATCHES THIS ADS2 REPORT"
           T "SPURIOUS ADS2 REPORT SUSPECTED" T
"SENTINEL DECISION AID CANNOT PROCESS THIS REPORT"
           T T
  "Now running a check on all past events . . ."
           T)
  (EVENTCHECK))
 (T

     (* If a match is made between an ADS1 EVENT property list and the ADS2 REPORT, the EVENT property list is marked by deleting P1TIME and preparations are made for Confidence processing and running the ADS2 Confidence rules et)

  (REMPROP EVENTMATCH (QUOTE P1TIME))
  (PUTPROP EVENTMATCH (QUOTE P2TIME)
           P2TIME)
  (PUTPROP EVENTMATCH (QUOTE P2HEADING)
           P2HEADING)
  (PutValue MATCHBoard (QUOTE LAUNCHSITE)
            (GETPROP EVENTMATCH (QUOTE P1LS)))
  (PutValue MATCHBoard (QUOTE P2LS)
            (GETPROP EVENTMATCH (QUOTE P1LS)))
  (PutValue MATCHBoard (QUOTE RPTSENSORCAP)
            SENSORCAP)
  (PutValue MATCHBoard (QUOTE OUTAGEREASON)
            REASONOUT)
  (PutValue MATCHBoard (QUOTE LSHEADING)
            P2HEADING)
  (PutValue MATCHBoard (QUOTE LSBSS)
            P2BSS)
  (RunRS (QUOTE SensorCheck))
  (RunRS (QUOTE DetectCheck))
  (RunRS (QUOTE ADS2Confidence])
(ADS2HANDLER)
78←
NIL
78←
NIL
78←
NIL
78←
NIL
78←
NIL
78←
NIL
78←(REDO 1)
```

using $BoardVoting

RuleSet BoardVoting

```
Last Edited: 3-Oct-85 18:30:33 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:  ;
Control Structure: DOALL;
********


(* BoardVoting checks every Board's EXPNUMBER. When the  EX
PNUMBER = 3 or 4, HIGHVOTE incremented by 1. LOWVOTE increm
ented when EXPNUMBER = 1 or 2. SYSBoard and IntelBoard are
special cases since they could be empty= 0.)

THEN (SETQ LOWVOTE 0)
     (SETQ HIGHVOTE 0);

(* No news is not necessarily good news, an empty Intel-  B
oard yields no information, so vote may be off by one.)
IF IntelBoard:EXPNUMBER = 0
THEN (SETQ LOWVOTE (SUB1 LOWVOTE));

IF   IntelBoard:EXPNUMBER <= 2
THEN (SETQ LOWVOTE (PLUS LOWVOTE 1));

IF IntelBoard:EXPNUMBER > 2
THEN (SETQ HIGHVOTE (PLUS HIGHVOTE 1));

(* With SYSBoard, Silence is held to be passive evidence of
 a well functioning system especially when we received our
satellite reports. 0 is a HIGHVOTE and effects ofgeneral ru
le counteracted.)

IF SYSBoard:EXPNUMBER = 0
THEN (SETQ LOWVOTE (SUB1 LOWVOTE))
     (SETQ HIGHVOTE (PLUS HIGHVOTE 1));

IF SYSBoard:EXPNUMBER <= 2
THEN (SETQ LOWVOTE (PLUS LOWVOTE 1));

IF SYSBoard:EXPNUMBER > 2
THEN (SETQ HIGHVOTE (PLUS HIGHVOTE 1));

IF TRENDBoard:EXPNUMBER <= 2
THEN (SETQ LOWVOTE (PLUS LOWVOTE 1));

IF TRENDBoard:EXPNUMBER > 2
THEN (SETQ HIGHVOTE (PLUS HIGHVOTE 1));
```

```
THEN (SETQ LOWVOTE (PLUS LOWVOTE 1));

IF TRENDBoard:EXPNUMBER > 2
THEN (SETQ HIGHVOTE (PLUS HIGHVOTE 1));

IF DetectBoard:EXPNUMBER <= 2
THEN (SETQ LOWVOTE (PLUS LOWVOTE 1));

IF DetectBoard:EXPNUMBER > 2
THEN (SETQ HIGHVOTE (PLUS HIGHVOTE 1));

IF MATCHBoard:EXPNUMBER <= 2
THEN (SETQ LOWVOTE (PLUS LOWVOTE 1));

IF MATCHBoard:EXPNUMBER > 2
THEN (SETQ HIGHVOTE (PLUS HIGHVOTE 1));
#&(RuleSetSource "G↑U0.zI[:.n59.70")
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←(REDO 1)
```

```
(BSSCALL
  [LAMBDA (LAUNCHSITE)                    (* AMRL " 5-Jun-85 14:08")

                    (* BSSCALL: This function takes BSSLIST and the RANGE informati
on of each BSS site, to compute a heading for the NORTH, SOUTH, EAST and WE
ST detection points from a launchsite to each BSS site.)


                                                (* INPUTS: BSSCALL uses LAUN
CHSITE and BSSLIST)                          (* OUTPUTS: HEADLIST provide to S
CHEDULER. BSSCALL also calls on POLAR with LNCHCOORD)

                    (* NOTE: The HEADLIST stores only two values the MAXPT which is
 the greatest angle that can be detected from that BSS site to the launchsite and t
he MINPT which is the smallest angle that can be detected.)


    (PROG (X Y BSSLONG BSSLAT BSSRANGE DETECTLIST EACHLIST
               BSSPOINT NORTHLAT SOUTHLAT NORTHEAD SOUTHEAD
               EASTLONG WESTLONG EASTHEAD)
                                                (* Set up the list and get Launch
site info from the property list)
          (SETQ DETECTLIST BSSLIST)
          (SETQ X (GETPROP LAUNCHSITE (QUOTE LAT)))
          (SETQ Y (GETPROP LAUNCHSITE (QUOTE LONG)))
          (SETQ LNCHCOORD (LIST X Y))
          [SETQ HEADLIST
            (LIST (CONS LAUNCHSITE (LIST (QUOTE MAXPT)
                                         (QUOTE MINPT]
                                                (* For each BSS site on the BSS
LIST; compute the heading information to each detection point. Assume circular c
overage pattern)
      LOOP(COND
            ((NULL (CAR DETECTLIST))
              (RETURN)))
          (SETQ BSSPOINT (CAR DETECTLIST))
          (SETQ BSSLAT (GETPROP BSSPOINT (QUOTE LAT)))
          (SETQ BSSLONG (GETPROP BSSPOINT (QUOTE LONG)))
          (SETQ BSSRANGE (FQUOTIENT (GETPROP BSSPOINT
                                             (QUOTE RANGE))
                                    (QUOTE 111)))
          (SETQ NORTHLAT (DIFFERENCE BSSLAT BSSRANGE))
          (POLAR (LIST NORTHLAT BSSLONG)
                 LNCHCOORD)
          (SETQ NORTHEAD HEADING)
          (SETQ SOUTHLAT (PLUS BSSLAT BSSRANGE))
          (POLAR (LIST SOUTHLAT BSSLONG)
                 LNCHCOORD)
          (SETQ SOUTHEAD HEADING)
          (SETQ EASTLONG (DIFFERENCE BSSLONG BSSRANGE))
          (POLAR (LIST BSSLAT EASTLONG)
                 LNCHCOORD)
```

```
        (SETQ HEADLIST
          (LIST (CONS LAUNCHSITE (LIST (QUOTE MAXPT)
                                       (QUOTE MINPT]
                                             (* For each BSS site on the BSS
LIST; compute the heading information to each detection point, Assume circular c
overage pattern)
      LOOP(COND
            ((NULL (CAR DETECTLIST))
              (RETURN)))
          (SETQ BSSPOINT (CAR DETECTLIST))
          (SETQ BSSLAT (GETPROP BSSPOINT (QUOTE LAT)))
          (SETQ BSSLONG (GETPROP BSSPOINT (QUOTE LONG)))
          (SETQ BSSRANGE (FQUOTIENT (GETPROP BSSPOINT
                                             (QUOTE RANGE))
                                    (QUOTE 111)))
          (SETQ NORTHLAT (DIFFERENCE BSSLAT BSSRANGE))
          (POLAR (LIST NORTHLAT BSSLONG)
                 LNCHCOORD)
          (SETQ NORTHEAD HEADING)
          (SETQ SOUTHLAT (PLUS BSSLAT BSSRANGE))
          (POLAR (LIST SOUTHLAT BSSLONG)
                 LNCHCOORD)
          (SETQ SOUTHEAD HEADING)
          (SETQ EASTLONG (DIFFERENCE BSSLONG BSSRANGE))
          (POLAR (LIST BSSLAT EASTLONG)
                 LNCHCOORD)
          (SETQ EASTHEAD HEADING)
          (SETQ WESTLONG (PLUS BSSLONG BSSRANGE))
          (POLAR (LIST BSSLAT WESTLONG)
                 LNCHCOORD)          (* Now we will take the min and
max headings of the four georgraphic points we have computed.)
          (SETQ MAXPT (FIX (FMAX NORTHEAD SOUTHEAD
                                 EASTHEAD HEADING)))
          (SETQ MINPT (FIX (FMIN NORTHEAD SOUTHEAD
                                 EASTHEAD HEADING)))
          (SETQ EACHLIST (LIST BSSPOINT MAXPT MINPT))
          (DOCOLLECT EACHLIST HEADLIST)
          (SETQ DETECTLIST (CDR DETECTLIST))
          (GO LOOP))
     (PUTPROP LAUNCHSITE (QUOTE BSSTAG)
              HEADLIST])
(BSSCALL)
90←
NIL
90←
NIL
90←
NIL
90←
NIL
90←(REDO 1)
```

```
(BSSEVENTCHECK
  [LAMBDA (MESSAGE)                          (* edited: " 8-Oct-85 14:50")
    (PROG (MATCHLIST MATCHPOINT MATCHTIME TIMEDIFF
                     OLDTIMEDIFF READ HEADINGDIFF
                     OLDHEADDIFF OLDHEADING)

          (* BSSEVENTCHECK: if the first attempt at matching a BSS event fai
ls, BSSEVENTCHECK makes a best guess and informs the user. The user may sele
ct the best guess or any remaining EVENT that has gone through the ADS-2 stage
of processing. Special case when P2HEADING = (QUOTE XXX) we can only use t
ime and the BSSTAG of the launcher to verify BSS detection.)


          (SETQ SENSOR (MKATOM (CONCAT (QUOTE TGT0)
                                       PSSITENUM)))
          (SETQ EVENTMATCH NIL)

          (* Best guess effort is the EVENT that results in combined smallest d
ifference between HEADING and TIME, found in the EVENTLIST, and the HEADING
 and TIME contained in the BSS messsage.)


          (PROG (BSSTEST MAXPT MINPT MARK)
                (SETQ MATCHLIST EVENTLIST)
                (SETQ OLDTIMEDIFF 500)
                (SETQ OLDHEADDIFF 70)
            LOOP(SETQ MATCHPOINT (CAR MATCHLIST))
                (SETQ MARK (QUOTE NIL))
                (COND
                  ((NULL MATCHPOINT)
                    (RETURN)))
                (SETQ MATCHTIME (GETPROP MATCHPOINT
                                         (QUOTE P2TIME)))
                [COND
                  ((NULL MATCHTIME)
                    (SETQ TIMEDIFF 501))
                  (T (SETQ TIMEDIFF
                       (ABS (DIFFERENCE PRESENTIME
                                        MATCHTIME]
                (SETQ OLDHEADING (GETPROP MATCHPOINT
                                          (QUOTE P2HEADING)
                                          ))
                [COND
                  ((EQP OLDHEADING (QUOTE XXX))
                    (SETQ OLDHEADING (QUOTE NIL]
                [COND
                  ((NULL OLDHEADING)
                    (SETQ HEADINGDIFF 71)
                    (SETQ OLDHEADING 0))
                  (T (SETQ HEADINGDIFF
```

```
            (SETQ OLDHEADING 0))
          (T (SETQ HEADINGDIFF
                (ABS (DIFFERENCE P3HEADING
                                OLDHEADING]
```

(* The following insures that the HEADING we are matching does fall in the range of detection BSSTAG of the launchsite. Therefore, if EVENT1 from LS6 has a heading of 230 then it can be detected by BSS5 ONLY if 230 lies between MAX and MIN angles of coverage. + and -.5 added to MAXPT and MINPT to insure that HEADINGs that are equal to MAXPT or MINPT will not be disqualifies.)

```
        [SETQ BSSTEST
          (ASSOC SENSOR
                (GETPROP (GETPROP MATCHPOINT
                                  (QUOTE P1LS))
                         (QUOTE BSSTAG]
        (SETQ MAXPT (CADR BSSTEST))
        (SETQ MINPT (CADDR BSSTEST))
        [COND
          ((AND (LESSP P3HEADING (PLUS MAXPT .5))
                (GREATERP P3HEADING
                          (PLUS MINPT -.5)))
            (SETQ MARK (QUOTE T]
```

(* This is the key condition for best guess; BSS coverage must be able to detect this event , MARK=T and then a combination of HEADING and TIME differences. Note that if one of the EVENTS matches either the TIME or HEADING, then that will be the best guess.)

```
        [COND
          ((EQP MARK (QUOTE T))
            (COND
              ((LESSP HEADINGDIFF OLDHEADDIFF)
                (COND
                  ((LESSP TIMEDIFF OLDTIMEDIFF)
                    (SETQ EVENTMATCH MATCHPOINT)
                    (SETQ OLDHEADIFF HEADINGDIFF)
                    (SETQ OLDTIMEDIFF TIMEDIFF]
        (SETQ MATCHLIST (CDR MATCHLIST))
        (GO LOOP))
```

(* Having found a best match, the following permits the user to select the best match first by offering him the SENTINEL solution than permitting him to pick his own.)

```
    [COND
      ((NOT (NULL EVENTMATCH))
        (printout T "Best Attempt at MATCH is: "
                  EVENTMATCH T T "EVENT HEADING: "
```

```
           ((NOT (NULL EVENTMATCH))
             (printout T "Best Attempt at MATCH is: "
                       EVENTMATCH T T "EVENT HEADING: "
                       (GETPROP EVENTMATCH (QUOTE
                                              P2HEADING))
                       T "Reported BSS HEADING: "
                       P3HEADING T
                    "Expected BSS MSG arrival time: "
                       (GETPROP EVENTMATCH (QUOTE P2TIME))
                       T "ACTUAL BSS MSG arrival time: "
                       P3TIME T T T
"If you concur with this event being matched type T in resp
onse, otherwise type NIL."
                       T)
             (SETQ READ (RATOM NIL NIL))
             (COND
               ((EQP READ (QUOTE T))
                 (RETURN (printout T
                        "Blackboards are being filled."
                                     T)
                          (REMPROP EVENTMATCH
                                   (QUOTE P2TIME))
                          (SETQ OLDHEADING
                            (GETPROP EVENTMATCH
                                     (QUOTE P2HEADING)))
                          (PUTPROP EVENTMATCH
                                   (QUOTE P2HEADING)
                                   P3HEADING)
                          (PutValue MATCHBoard
                                    (QUOTE OUTAGEREASON)
                                    (GETPROP SENSOR
                                             (QUOTE REASON)
                                             ))
                          (PutValue MATCHBoard
                                    (QUOTE RPTSENSORCAP)
                                    (GETPROP SENSOR
                                             (QUOTE
                                            CAPABILITY)))
                          (PutValue MATCHBoard
                                    (QUOTE P2LS)
                                    (GETPROP EVENTMATCH
                                             (QUOTE P1LS)))
                          (PutValue MATCHBoard
                                    (QUOTE LSHEADING)
                                    (GETPROP EVENTMATCH
                                             (QUOTE
                                             P2HEADING)))
                          (RunRS (QUOTE SensorCheck))
                          (RunRS (QUOTE DetectCheck))
                          (RunRS (QUOTE BSSTrendCheck))
```

```
(PUTPROP EVENTMATCH
         (QUOTE P2HEADING)
         OLDHEADING]
```

(* IF all else fails, BSSEVENTCHECK goes down the list of EVENTS one by one and shows them to the user and asks if he wishes to match them. Only EVENTS that are off limits are those which do not have a P1TIME and thus have been through ADS-2 Processing.)

```
(PROG (READ MARK BSSTEST MAXPT MINPT)
      (printout T T
"BSS EVENT CHECK will check through all appropriate events
to match BSS REPORT."
                T)
      (SETQ MATCHLIST EVENTLIST)
  LOOP(SETQ MARK (QUOTE T))
      [COND
        ((NULL (CAR MATCHLIST))
          (RETURN (printout T
                     "EVENT LIST EXHAUSTED."
                            T]
      (SETQ MATCHPOINT (CAR MATCHLIST))
      (COND
        ([NOT (NULL (GETPROP MATCHPOINT
                             (QUOTE P1TIME]
          (SETQ MATCHLIST (CDR MATCHLIST))
          (GO LOOP)))
```

(* Above COND eliminates those EVENTS which have only been matched to an ADS-1 Report (hence P1TIME still has a value), Next COND eliminates those EVENTS which have already been matched to an BSS report, for those reports have no P2TIME)

```
      (COND
        ((NULL (GETPROP MATCHPOINT (QUOTE P2TIME)
                        ))
          (SETQ MATCHLIST (CDR MATCHLIST))
          (GO LOOP)))
      [SETQ BSSTEST
        (ASSOC SENSOR
               (GETPROP (GETPROP MATCHPOINT
                                 (QUOTE P1LS))
                        (QUOTE BSSTAG]
      (SETQ MAXPT (CADR BSSTEST))
      (SETQ MINPT (CADDR BSSTEST))
      (printout T "ATTEMPT to MATCH: "
                MATCHPOINT T T
 "This BSS site can detect missiles at these HEADINGS"
                T
```

```
                    "Expected BSS MSG arrival time: "
                         (GETPROP MATCHPOINT
                                  (QUOTE P2TIME))
                         T
                      "ACTUAL BSS MSG arrival time: "
                         P3TIME T T T
          "IF you wish to match this event, type T"
                         T
      "Otherwise type NIL to examine other events,"
                         T
                  "OR type STOP if you wish to exit."
                         T)        (* The following is the user input.
 NIL continues the LOOP, T matches the EVENT and resumes processing, STOP te
rminates the process.)
               (SETQ READ (RATOM NIL NIL))
               (COND
                 ((EQP READ (QUOTE NIL))
                   (SETQ MATCHLIST (CDR MATCHLIST))
                   (GO LOOP)))
               (COND
                 ((EQP READ (QUOTE T))
                   (SETQ EVENTMATCH MATCHPOINT)
                   (REMPROP EVENTMATCH (QUOTE P2TIME))
                   (printout T
                      "Blackboards are being filled."
                            T)
                   (SETQ OLDHEADING (GETPROP
                       EVENTMATCH
                       (QUOTE P2HEADING)))
                   (PUTPROP EVENTMATCH (QUOTE P2HEADING)
                            P3HEADING)
                   (PutValue MATCHBoard (QUOTE
                                          OUTAGEREASON)
                             (GETPROP SENSOR
                                      (QUOTE REASON)))
                   (PutValue MATCHBoard (QUOTE
                                          RPTSENSORCAP)
                             (GETPROP SENSOR
                                      (QUOTE CAPABILITY)))
                   (PutValue MATCHBoard (QUOTE P2LS)
                             (GETPROP EVENTMATCH
                                      (QUOTE P1LS)))
                   (PutValue MATCHBoard (QUOTE LSHEADING)
                             (GETPROP EVENTMATCH
                                      (QUOTE P2HEADING)))
                   (RunRS (QUOTE SensorCheck))
                   (RunRS (QUOTE DetectCheck))
                   (RunRS (QUOTE BSSTrendCheck))
                   (PUTPROP EVENTMATCH (QUOTE P2HEADING)
                            OLDHEADING]))
```

```
(BSSHANDLER
  [LAMBDA (MESSAGE)                          (* edited: " 8-Oct-85 14:30")
    (PROG (BSSMSG MATCHLIST MATCHPOINT MATCHTIME TIMEDIFF
                  OLDTIMEDIFF HEADINGDIFF OLDHEADDIFF
                  OLDHEADING MARK BSSTEST MAXPT MINPT)

          (* BSSHANDLER first decomposes the BSS message into its constitu
ent parts and then matches the BSS message with the appropriate EVENT from the
 EVENTLIST. Once matched, EVENTMATCH is the key to further BSS processing. I
f not matched, EVENTCHECK is invoked.)


          (SETQ BSSMSG MESSAGE)
          (SETQ P3SITENUM (CAAADR (CADAR BSSMSG)))
          (SETQ P3TIME (CADAAR (CDADAR BSSMSG)))
          (SETQ P3OBJNUM (CADADR (CADAR BSSMSG)))
          (SETQ P3HEADING (CADDR (CADAR BSSMSG)))
          (SETQ P3STABILITY (CADDAR BSSMSG))
          (SETQ P3IMPACTTIME (CAADR BSSMSG))
          (SETQ P3SENSORRATING (CADDR BSSMSG))
          (SETQ SENSOR (MKATOM (CONCAT (QUOTE TGT0)
                                       P3SITENUM)))
          (printout T "Detecting BSS SITE: " P3SITENUM T
                    "TIME of REPORT: "
                    P3TIME T
                    "Number of Objects detected: "
                    P3OBJNUM T "DETECTED HEADING: "
                    P3HEADING T "Trajectory Stability: "
                    P3STABILITY T
                    " Expected IMPACT TIME: "
                    P3IMPACTTIME T "BSS Site Confidence: "
                    P3SENSORRATING T)

          (* MATCHING part is here: In order to achieve a MATCHED EVENT f
rom the EVENTLIST, the P2HEADING must be as close as possible to the BSS det
ected HEADING and the report must be within 3 minutes of the estimated time. HE
ADING cannot vary more than 5 degrees, the maximum manuvering range of the T
YPE II missile.)


          (SETQ MATCHLIST EVENTLIST)
          (SETQ OLDHEADDIFF 6)
          (SETQ EVENTMATCH NIL)
      LOOP1
          (SETQ MATCHPOINT (CAR MATCHLIST))
          (COND
            ((NULL MATCHPOINT)
              (RETURN)))
          (SETQ OLDHEADING (GETPROP MATCHPOINT
                                    (QUOTE P2HEADING)))
                                    (* Special case: where P2HEADI
```

```
                                        (QUOTE P2HEADING)))
                                      (* Special case; where P2HEADI
NG from EVENTLIST was not discernible.)
          [COND
            ((EQP OLDHEADING (QUOTE XXX))
              (SETQ OLDHEADING (QUOTE NIL]
          [COND
            ((NULL OLDHEADING)
              (SETQ HEADINGDIFF 7))
            (T (SETQ HEADINGDIFF (ABS (DIFFERENCE
                                            P3HEADING
                                           OLDHEADING]

          (* The following is where the P3HEADING reported by the BSS mess
age is compared to the MAXPT and MINPT zones of coverage of the BSS site. Onl
y if P3HEADING would normally fall in the zone of coverage of this BSS site, will it
 be considered for a MATCH. MARK=T indicates compatibility.)


          (SETQ MARK (QUOTE NIL))
          [SETQ BSSTEST (ASSOC SENSOR
                               (GETPROP
                                 (GETPROP MATCHPOINT
                                          (QUOTE P1LS))
                                 (QUOTE BSSTAG]
          (SETQ MAXPT (CADR BSSTEST))
          (SETQ MINPT (CADDR BSSTEST))
          [COND
            ((AND (LESSP P3HEADING (PLUS MAXPT .5))
                  (GREATERP P3HEADING (PLUS MINPT -.5)))
              (SETQ MARK (QUOTE T]
          (SETQ MATCHTIME (GETPROP MATCHPOINT
                                   (QUOTE P2TIME)))
          [COND
            ((NULL MATCHTIME)
              (SETQ TIMEDIFF 202))
            (T (SETQ TIMEDIFF (ABS (DIFFERENCE PRESENTIME
                                               MATCHTIME]

          (* This is the key COND in the program. If the BSS site covers this h
eading from the EVENT launchsite MARK='T, and if TIME for the BSS report does
 not exceed 2 minutes 501 then the EVENT which meets all these conditions and
whose HEADING is closest to P2HEADING from the EVENTLIST will be EVENTMAT
CH.)


          [COND
            ((EQP MARK (QUOTE T))
              (COND
                ((LESSP HEADINGDIFF OLDHEADDIFF)
                  (COND
                    ((LESSP TIMEDIFF 201)
```

```
                    ((LESSP HEADINGDIFF OLDHEADDIFF)
                      (COND
                        ((LESSP TIMEDIFF 201)
                          (SETQ EVENTMATCH MATCHPOINT)
                          (SETQ OLDHEADIFF HEADINGDIFF]
              (SETQ MATCHLIST (CDR MATCHLIST))
              (GO LOOP1))                    (* RESULTS are announced, if fir
st match is unsuccessful BSSEVENTCHECK is called, The blackboards are filled a
gain.)
    (COND
      ((NULL EVENTMATCH)
        (printout T T
       "BSS Report not matched on first attempt . . ."
                  T)
        (BSSEVENTCHECK MESSAGE))
      (T (printout T T "BSS Report Matched to: "
                   EVENTMATCH T T)
         (printout T "Blackboards are being filled. " T)
         (SETQ OLDHEADING (GETPROP EVENTMATCH
                                   (QUOTE P2HEADING)))
         (PUTPROP EVENTMATCH (QUOTE P2HEADING)
                  P3HEADING)
         (REMPROP EVENTMATCH (QUOTE P2TIME))
         (PutValue MATCHBoard (QUOTE OUTAGEREASON)
                   (GETPROP SENSOR (QUOTE REASON)))
         (PutValue MATCHBoard (QUOTE RPTSENSORCAP)
                   (GETPROP SENSOR (QUOTE CAPABILITY)))
         (PutValue MATCHBoard (QUOTE P2LS)
                   (GETPROP EVENTMATCH (QUOTE P1LS)))
         (PutValue MATCHBoard (QUOTE LSHEADING)
                   (GETPROP EVENTMATCH (QUOTE P2HEADING)))
         (RunRS (QUOTE SensorCheck))
         (RunRS (QUOTE DetectCheck))
         (RunRS (QUOTE BSSTrendCheck))
         (PUTPROP EVENTMATCH (QUOTE P2HEADING)
                  OLDHEADING])
(BSSHANDLER)
97←
NIL
97←
NIL
97←
NIL
97←
NIL
97←
NIL
97←
NIL
97←(REDO 100)
```

using $BSSHOSTILE

RuleSet BSSHOSTILE

```
Last Edited: 10-Oct-85 17:05:41 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   MARK READ OUTVAR XVAR YVAR;
Control Structure: DOALL;
********



(* BSSHOSTILE processes the BSS message for those events wh
ich have been identified as HOSTILE. First, BSSHOSTILE   co
nsults BoardVoting to determine when confidence level   rev
ision is appropriate, and then consults BSSBoard quite heav
ily to determine when hypothesis revision to other cases su
ch as TEST, UNRESOLVED or UNKNOWN is in order. BSSHOSTILE t
hen fills out the BSS report for the MWO and permits explan
ation using Explain and Explain2.)

THEN (RunRS 'BoardVoting);

IF HIGHVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

IF HIGHVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF LOWVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF LOWVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

(* Hypothesis revision takes place in the following.)

IF BSSBoard:EXPNUMBER <= 2
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
   HYPOTRANSFER = 'NIL
THEN (SETQ HYPOTRANSFER 'T)
     (RunRS 'BSSUNRESOLVED) (STOP);

IF BSSBoard:EXPNUMBER <= 2
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
   HYPOTRANSFER = 'NIL
THEN (SETQ HYPOTRANSFER 'T)
```

```
THEN (SETQ HYPOTRANSFER 'T)
     (RunRS 'BSSUNRESOLVED) (STOP);

IF HYPOTRANSFER = 'NIL
   BSSBoard:EXPNUMBER = 1
THEN (SETQ HYPOTRANSFER 'T)
     (RunRS 'BSSTEST) (STOP);

(* Unless THREAT is specifically disproved, threat is assum
ed to be YES since hypothesis is HOSTILE.)

THEN OUTVAR←'YES;

IF BSSBoard:EXPNUMBER < 2
THEN OUTVAR←'NO;

(* The different lists are combined for both HIGHVALUE and
 other targets.)

IF TRENDBoard:WILLNUMBER >= 1
THEN (PutValue BSSBoard 'WILLDETECT (APPEND BSSBoard:WILLDE
TECT  TRENDBoard:WILLDETECT));

IF TRENDBoard:CLOSENUMBER >= 1
THEN (PutValue BSSBoard 'CLOSEDETECT (APPEND BSSBoard:CLOSE
DETECT  TRENDBoard:CLOSEDETECT));

IF TRENDBoard:FARNUMBER >= 1
THEN (PutValue BSSBoard 'FARDETECT (APPEND BSSBoard:FARDETE
CT   TRENDBoard:FARDETECT));

(* Output Section)

THEN (PRINTOUT T T "EVENT Considered: HOSTILE" T T T "    B
SS  " EVENTMATCH T T "1. DETECTED: " P3TIME T "2. BSS: "  P
3SITENUM T "3. THREAT: " OUTVAR  T "4. LAUNCHER TYPE: " MAT
CHBoard:LAUNCHERTYPE T "5. OBJECTS TRACKED: "           P3
OBJNUM T)  MARK←'NIL;

IF BSSBoard:WILLDETECT ~= 'NIL
THEN MARK←'T
     (PRINTOUT T "6. VERY LIKELY TARGETS: " (CONVERT BSSBoa
rd:WILLDETECT) T);

IF MARK = 'NIL
   BSSBoard:CLOSEDETECT ~= 'NIL
THEN MARK←'T
     (PRINTOUT T "6. LIKELY TARGETS: " (CONVERT BSSBoard:CL
OSEDETECT) T);
```

```
IF MARK = 'NIL
   BSSBoard:CLOSEDETECT ~= 'NIL
THEN MARK←'T
     (PRINTOUT T "6. LIKELY TARGETS: " (CONVERT BSSBoard:CL
OSEDETECT) T);

IF MARK = 'NIL
   BSSBoard:FARDETECT ~= 'NIL
THEN MARK←'T
     (PRINTOUT T "6. POSSIBLE TARGETS: " (CONVERT BSSBoard:
FARDETECT) T);

IF MARK = 'NIL
THEN (PRINTOUT T "6. POSSIBLE TARGETS: NONE" T);

THEN (PRINTOUT T "7. EST. IMPACT TIME: " P3IMPACTTIME T "8.
 CONFIDENCE: " (GETPROP EVENTMATCH 'P2CONFIDENCE) T T)
  (RunRS 'Speculate);

IF EXPLAINBoard:COMMENT ~= 'NIL
THEN (EVAL EXPLAINBoard:COMMENT);

THEN (PRINTOUT T T "If you wish further explanation type T,
" T "Otherwise type NIL to proceed." T)
    (SETQ READ (RATOM NIL NIL));

IF READ = 'T
THEN (SETQ FACTORLIST (LIST BSSBoard:EXPNUMBER 'BSSBoard  T
RENDBoard:EXPNUMBER 'TRENDBoard MATCHBoard:EXPNUMBER 'MATCH
Board IntelBoard:EXPNUMBER 'IntelBoard SYSBoard:EXPNUMBER '
SYSBoard));

IF READ = 'T
   OUTVAR = 'NO
THEN (RunRS 'LOWRanking) (RunRS 'Explain)(STOP);

IF READ = 'T
   OUTVAR = 'YES
THEN (RunRS 'HIGHRanking) (RunRS 'Explain);
#&(RuleSetSource "ODU0.zI[:.Eb:.3")
26←
NIL
26←
NIL
26←
NIL
26←
NIL
26←(REDO 100)
```

```
using $BSSensorCheck
```
RuleSet BSSensorCheck

```
Last Edited:  3-Oct-85 18:48:52 by
WorkSpace Class: Targets;
Compiler Options:  ;
Temporary Vars:   P3SENSORCAP P3SENSORCONF;
Control Structure: DOALL;
********



(* BSSensorCheck aids BSSTrendCheck in evaluating the     T
rend of BSS evidence. P3SENSOR is computed by comparing the
 BSS site confidence and capability.)

THEN (SETQ P3SENSORCAP (GETPROP SENSOR 'CAPABILITY))
     (SETQ P3SENSOR 0);

(* First P3SENSORRATING is mapped to the categories of    H
IGH, MEDUIM, or LOW.)

IF P3SENSORRATING >= 7
THEN  P3SENSORCONF←'HIGH;

IF P3SENSORRATING >= 4
   P3SENSORRATING < 7
THEN P3SENSORCONF←'MEDUIM;

IF P3SENSORRATING < 4
THEN P3SENSORCONF←'LOW;

(* Now P3SENSOR can be determined.)
IF P3SENSORCAP = 'FULL
   P3SENSORCONF = 'HIGH
THEN (SETQ P3SENSOR 9);

IF P3SENSORCAP = 'FULL
   P3SENSORCONF = 'MEDUIM
THEN (SETQ P3SENSOR 8);

IF P3SENSORCAP = 'PARTIAL
   P3SENSORCONF = 'HIGH
THEN (SETQ P3SENSOR 7);

IF P3SENSORCAP = 'FULL
   P3SENSORCONF = 'LOW
THEN (SETQ P3SENSOR 6);

IF P3SENSORCAP = 'PARTIAL
   P3SENSORCONF = 'MEDUIM
```

```
IF P3SENSORCAP = 'PARTIAL
   P3SENSORCONF = 'MEDUIM
THEN (SETQ P3SENSOR 5);

IF P3SENSORCAP = 'ZERO
   P3SENSORCONF = 'HIGH
THEN (SETQ P3SENSOR 4);

IF P3SENSORCAP = 'PARTIAL
   P3SENSORCONF = 'LOW
THEN (SETQ P3SENSOR 3);

IF P3SENSORCAP = 'ZERO
   P3SENSORCONF = 'MEDUIM
THEN (SETQ P3SENSOR 2);

IF P3SENSORCAP = 'ZERO
   P3SENSORCONF = 'LOW
THEN (SETQ P3SENSOR 1);

THEN (PRINTOUT T "this is P3SENSORRATING: " P3SENSORRATING
T "This is P3SENSORCONF " P3SENSORCONF T "thsi is P3SESNORC
AP " P3SENSORCAP T "this is P3SENSOR " P3SENSOR T );
#&(RuleSetSource "3WU0.zI[:.<;:.7")
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←
NIL
32←(REDO 100)
```

using $BSSTEST

RuleSet BSSTEST

```
Last Edited: 8-Oct-85 14:58:59 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   MARK READ OUTVAR XVAR YVAR;
Control Structure: DOALL;
********


(* BSSTEST processes the BSS message for those events which
 have been identified as TEST. First, BSSTEST consults Boar
dVoting to determine when confidence level revision is appr
opriate, and then consults BSSBoard quite heavily to determ
ine when hypothesis revision to other cases such as   HOSTI
LE, UNRESOLVED or UNKNOWN is in order. BSSTEST then  fills
out the BSS report for the MWO and permits          explana
tion using Explain and Explain2.)

THEN (RunRS 'BoardVoting);

IF HIGHVOTE >= 3
  (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

IF HIGHVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF LOWVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF LOWVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

(* Having accomplished confidence revision when necessary,
hypothesis revision can take place below.)

IF BSSBoard:EXPNUMBER <= 2
    (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
    HYPOTRANSFER = 'NIL
THEN (SETQ HYPOTRANSFER 'T)
     (RunRS 'BSSUNRESOLVED) (STOP);

IF BSSBoard:EXPNUMBER <= 2
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
   HYPOTRANSFER = 'NIL
```

```
   HYPOTRANSFER = 'NIL
THEN (SETQ HYPOTRANSFER 'T)
     (RunRS 'BSSUNRESOLVED) (STOP);

IF HYPOTRANSFER = 'NIL
   BSSBoard:EXPNUMBER > 2
THEN (SETQ HYPOTRANSFER 'T)
     (RunRS 'BSSHOSTILE) (STOP);



(* The C3 THREAT and other target THREAT are combined toget
her into the LISTs on BSSBoard. THREAT set to NO first.)

 THEN OUTVAR←'NO;

IF BSSBoard:EXPNUMBER > 2
THEN OUTVAR←'YES;



IF TRENDBoard:WILLNUMBER >= 1
THEN (PutValue BSSBoard 'WILLDETECT (APPEND BSSBoard:WILLDE
TECT  TRENDBoard:WILLDETECT));

IF TRENDBoard:CLOSENUMBER >= 1
THEN (PutValue BSSBoard 'CLOSEDETECT (APPEND BSSBoard:CLOSE
DETECT  TRENDBoard:CLOSEDETECT));

IF TRENDBoard:FARNUMBER >= 1
THEN (PutValue BSSBoard 'FARDETECT (APPEND BSSBoard:FARDETE
CT TRENDBoard:FARDETECT));

 (* Output Section)

THEN (PRINTOUT T T "EVENT Considered: TEST" T T T "    BSS
 " EVENTMATCH T T "1. DETECTED: " P3TIME T "2. BSS: "  P3SI
TENUM T "3. THREAT: " OUTVAR  T "4. LAUNCHER TYPE: " MATCHB
oard:LAUNCHERTYPE T "5. OBJECTS TRACKED: "            P3OBJ
NUM T)  MARK←'NIL;

IF BSSBoard:WILLDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. VERY LIKELY TARGETS: "
(CONVERT BSSBoard:WILLDETECT) T);

IF MARK = 'NIL
    BSSBoard:CLOSEDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. LIKELY TARGETS: " (CONVERT
  BSSBoard:CLOSEDETECT) T);
```

```
THEN MARK←'T (PRINTOUT T "6. LIKELY TARGETS: " (CONVERT
  BSSBoard:CLOSEDETECT) T);

IF MARK = 'NIL
    BSSBoard:FARDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. POSSIBLE TARGETS: " (CONVERT
BSSBoard:FARDETECT) T);

IF MARK = 'NIL
THEN (PRINTOUT T "6. POSSIBLE TARGETS: NONE" T);

THEN (PRINTOUT T "7. EST. IMPACT TIME: " P3IMPACTTIME T "8.
 CONFIDENCE: " (GETPROP EVENTMATCH 'P2CONFIDENCE) T T) (Run
RS 'Speculate);

IF EXPLAINBoard:COMMENT ~= 'NIL
THEN (EVAL EXPLAINBoard:COMMENT);

THEN (PRINTOUT T T "If you wish further explanation type T,
" T "Otherwise type NIL to proceed." T) (SETQ READ (RATOM N
IL NIL));

IF READ = 'T
THEN (SETQ FACTORLIST (LIST BSSBoard:EXPNUMBER 'BSSBoard  T
RENDBoard:EXPNUMBER 'TRENDBoard MATCHBoard:EXPNUMBER 'MATCH
Board IntelBoard:EXPNUMBER 'IntelBoard SYSBoard:EXPNUMBER '
SYSBoard));

IF READ = 'T
   OUTVAR = 'NO
THEN (RunRS 'LOWRanking)
     (RunRS 'Explain) (STOP);

IF READ = 'T
   OUTVAR = 'YES
THEN (RunRS 'HIGHRanking) (RunRS 'Explain);
#&(RuleSetSource (10 . 49284))
13←
NIL
13←
NIL
13←
NIL
13←
NIL
13←
NIL
13←
NIL
13←(REDO 100)
```

using $BSSTGTCheck

RuleSet BSSTGTCheck

```
Last Edited: 8-Oct-85 14:47:41 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   TGTWILLDETECT TGTCLOSEDETECT TGTFARDETECT
 TLIST TPOINT TRANGE THEADING TARGET THEADIFF MARK TLIST RA
NGEMARK MAXRANGE LSTYPE RANGEDIFF;
Control Structure: WHILEALL;
Iteration Condition:   T;
********



(* BSSTGTCHECK is used by the BSS message processing to tak
e the FANCALL of the EVENT and place all of the targets in
three categories: WILLDETECT are those targets within 1 deg
ree of missile heading, CLOSEDETECT are those within 5 degr
ees, and FARDETECT are up to 10 degrees. Note that each tar
get will appear in only one of those lists. RANGE informati
on is used to reduce the FANCALL search requriements. Resul
ts are placed on the BSSBoard blackboard.)

{1!} THEN TLIST←(CDR TAGLIST)
     BSSBoard:WILLNUMBER←0
     BSSBoard:CLOSENUMBER←0
     BSSBoard:FARNUMBER←0
     BSSBoard:WILLDETECT←NIL
     BSSBoard:CLOSEDETECT←NIL
     BSSBoard:FARDETECT←NIL
     LSTYPE←MATCHBoard:LAUNCHERTYPE
     (SETQ HEADING MATCHBoard:LSHEADING);

IF (CAR TLIST) = 'NIL
THEN BSSBoard:WILLDETECT←TGTWILLDETECT
     BSSBoard:CLOSEDETECT←TGTCLOSEDETECT
     BSSBoard:FARDETECT←TGTFARDETECT
(PRINTOUT T "This is WILLDETECT: " TGTWILLDETECT T
    "This is CLOSEDETECT: " TGTCLOSEDETECT T
    "THIS  is FARDETECT: " TGTFARDETECT T
    "This is WILLNUMBER: " BSSBoard:WILLNUMBER T
    "this is CLOSENUMBER: " BSSBoard:CLOSENUMBER T
   "This is FARNUMBER: " BSSBoard:FARNUMBER T)
 (STOP);

THEN TPOINT←(CAR TLIST)
     TARGET←(CAR TPOINT)
     TRANGE←(CADDR TPOINT)
     THEADING←(CADR TPOINT)
     MARK←'T RANGEMARK←'T
```

```
(STOP);

THEN TPOINT←(CAR TLIST)
     TARGET←(CAR TPOINT)
     TRANGE←(CADDR TPOINT)
     THEADING←(CADR TPOINT)
     MARK←'T RANGEMARK←'T
     THEADIFF←(ABS (DIFFERENCE THEADING HEADING));

(* MAXIMUM range of each missile type is specified here)

IF LSTYPE='TYPE1
THEN MAXRANGE←1325;

IF LSTYPE = 'TYPE2
THEN MAXRANGE←2650;

THEN RANGEDIFF←(DIFFERENCE MAXRANGE TRANGE);

IF RANGEDIFF < 0
THEN RANGEMARK←'NIL;

IF THEADIFF <= 1
   RANGEMARK = 'T
THEN TGTWILLDETECT←(APPEND (LIST TARGET) TGTWILLDETECT)   M
ARK←'NIL
 BSSBoard:WILLNUMBER←(PLUS TRENDBoard:WILLNUMBER 1);

IF THEADIFF <= 5
   MARK = 'T
   RANGEMARK = 'T
THEN TGTCLOSEDETECT←(APPEND (LIST TARGET) TGTCLOSEDETECT) B
SSBoard:CLOSENUMBER←(PLUS TRENDBoard:CLOSENUMBER 1)   MARK←
'NIL;

IF THEADIFF <= 10
   MARK = 'T
   RANGEMARK = 'T
THEN TGTFARDETECT←(APPEND (LIST TARGET) TGTFARDETECT)
  BSSBoard:FARNUMBER←(PLUS TRENDBoard:FARNUMBER 1);

THEN TLIST←(CDR TLIST);
#&(RuleSetSource "SWU0.zI[:.<;:.5")
18←
NIL
18←
NIL
18←
NIL
18←(REDO 100)
```

```
using $BSSTrendCheck
RuleSet BSSTrendCheck
Last Edited: 9-Oct-85 14:03:24 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   FIRSTCONF SECONDCONF UPNUMBER MARK;
Control Structure: DOALL;
********



(* BSSTrendCheck weighs the evidence from 3 factors CONFIDE
NCE LEVELS, IR TYPE and SENSOR RATING from ADS-1 to the ADS
-2 Report, and to the BSS REPORT and attempts to discern wh
ether the Trend of evidence is going up or down.)

THEN UPNUMBER←0 ;

(* To make it easier to compare, first P1CONFIDENCE and P2C
ONFIDENCE from the EVENT LIST are mapped to numbers.)

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'EXTHIGH
THEN FIRSTCONF←5;

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'HIGH
 THEN FIRSTCONF←4;

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'MIXED
 THEN FIRSTCONF←3;

 IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'LOW
THEN FIRSTCONF←2;

 IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'EXTLOW
 THEN FIRSTCONF←1;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN SECONDCONF←5;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'MIXED
THEN SECONDCONF←3;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN SECONDCONF←4;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN SECONDCONF←2;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
THEN SECONDCONF←1;
```

```
 (* Here they are compared, in case of tie, TREND is UP.) I
F SECONDCONF >= FIRSTCONF
 THEN UPNUMBER←(PLUS UPNUMBER 1);

(* FIRSTCONF variable recycled: Here it is used with BSS lo
cal variables to verify TYPE1 or TYPE2 Missile data.)

THEN FIRSTCONF←'TYPE2;

IF P3STABILITY = 'Y
   P3OBJNUM < 2
   (GETPROP EVENTMATCH 'P1IRTYPE) ~= 'TYPE2
THEN  FIRSTCONF←'TYPE1;

 (* The Infrared Type TREND takes advantage of the typing o
f ADS-2 Feature Set into MATCH, MISMATCH, NOMATCH. Only way
 to get DOWN trend is to get MISMATCH or a MATCH on an Indi
stinct or Unknown ADS-1 reading.)

 IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'INDISTINCT
   (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN UPNUMBER←(SUB1 UPNUMBER);

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'UNKNOWN
   (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN UPNUMBER←(SUB1 UPNUMBER);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN UPNUMBER←(PLUS UPNUMBER 1);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'NOMATCH
THEN UPNUMBER←(PLUS UPNUMBER 1);

(* For BSS cases FIRSTCONF compared, when all 3 measurement
s occur, net effect=2, otherwise plus one.)

IF FIRSTCONF = 'TYPE1
   (GETPROP EVENTMATCH 'P1IRTYPE) = 'TYPE1
   (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN UPNUMBER←(PLUS UPNUMBER 1);

IF FIRSTCONF = 'TYPE2
   (GETPROP EVENTMATCH 'P1IRTYPE) = 'TYPE2
   (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN UPNUMBER←(PLUS UPNUMBER 1);

IF (GETPROP EVENTMATCH 'P2TYPE) ~= 'MISMATCH
THEN UPNUMBER←(PLUS UPNUMBER 1);
```

```
THEN UPNUMBER←(PLUS UPNUMBER 1);

(* SENSOR Ratings can be compared directly since they
are already numbers. TIE is an UP vote.)

 IF (GETPROP EVENTMATCH 'P2SENSOR) >=
    (GETPROP EVENTMATCH 'P1SENSOR)
THEN UPNUMBER←(PLUS UPNUMBER 1);

(* BSSensorCheck is run to compute a value for P3SENSOR)


THEN (RunRS 'BSSensorCheck)
  (PRINTOUT T "UPNUMBER before BSSensorCheck " UPNUMBER T "
This is P3Sensor : " P3SENSOR T);

IF P3SENSOR >= (GETPROP EVENTMATCH 'P2SENSOR)
THEN UPNUMBER←(PLUS UPNUMBER 1);

IF P3SENSOR >= (GETPROP EVENTMATCH 'P1SENSOR)
THEN UPNUMBER←(PLUS UPNUMBER 1);

THEN (PRINTOUT T "This is UPNUMBER " UPNUMBER T);

 (* Now the results are interpreted, COMMENT and EXPNUMBER
on the TRENDBoard are set.)

 IF UPNUMBER > 5
 THEN TRENDBoard:EXPNUMBER←4  TRENDBoard:COMMENT←'(PRINTOUT
 T "Trend of Evidence is: POSITIVELY UP" T);

 IF UPNUMBER >= 4
    UPNUMBER <= 5
 THEN TRENDBoard:EXPNUMBER←3 TRENDBoard:COMMENT←'(PRINTOUT
T "Trend of Evidence is: UP " T);

 IF UPNUMBER <= 4
     UPNUMBER >= 3
 THEN TRENDBoard:EXPNUMBER←2 TRENDBoard:COMMENT←'(PRINTOUT
T "Trend of Evidence is: DOWN " T);

 IF UPNUMBER < 3
 THEN TRENDBoard:EXPNUMBER←1  TRENDBoard:COMMENT←'(PRINTOUT
 T "Trend of Evidence is: POSITIVELY DOWN" T);

 THEN (EVAL TRENDBoard:COMMENT)
      (SETQ TAGLIST (GETPROP (GETPROP EVENTMATCH 'P1LS) 'FA
NCALL)) MARK←'NIL  FIRSTCONF←'NIL
      (RunRS 'BSSTGTCheck);

        .
```

```
nd FIRSTCONF flags are set for the COMMENTS.)

IF BSSBoard:WILLNUMBER >= 1
THEN BSSBoard:EXPNUMBER←4
     BSSBoard:COMMENT←'(PRINTOUT T T "Threat to US Targets:
 EXTREMELY LIKELY" T)    MARK←'T;

IF MARK = 'NIL
   BSSBoard:CLOSENUMBER >= 1
THEN BSSBoard:EXPNUMBER←3
     BSSBoard:COMMENT←'(PRINTOUT T T "Threat to US Targets:
 LIKELY" T) MARK←'T;

IF MARK = 'NIL
   BSSBoard:FARNUMBER >= 1
THEN BSSBoard:EXPNUMBER←2
     BSSBoard:COMMENT←'(PRINTOUT T T "Threat to US Targets:
 POSSIBLE" T)  MARK←'T;

IF MARK = 'NIL
THEN BSSBoard:EXPNUMBER←1
     BSSBoard:COMMENT←'(PRINTOUT T T "Threat to US Targets:
 NONE " T);

(* Since BSS event processing makes the DETECTBoard irrelav
ant the previous C3 target threat COMMENT previously usedwi
th DETECTBoard is reproduced for BSSBoard.)

IF TRENDBoard:WILLNUMBER >= 1
THEN MARK←'("Threat to C3 Nodes: VERY HIGH" T)
     FIRSTCONF←'T  BSSBoard:EXPNUMBER←4;

IF FIRSTCONF = 'NIL
   TRENDBoard:CLOSENUMBER >= 1
THEN MARK←'("Threat to C3 Nodes: HIGH " T)
     FIRSTCONF←'T;

(* HIGH or VERY HIGH C3 Node Threat will over ride the EXPN
UMBER set above since sometimes no or few US targets will b
e threatened but C3 Nodes are under heavy threat.)

IF FIRSTCONF = 'T
   BSSBoard:EXPNUMBER < 3
THEN BSSBoard:EXPNUMBER←3;

IF FIRSTCONF = 'NIL
   TRENDBoard:FARNUMBER >= 1
THEN MARK←'("Threat to C3 Nodes: LOW" T)
     FIRSTCONF←'T;
```

```
   TRENDBoard:PARNUMBER >= 1
THEN MARK←'("Threat to C3 Nodes: LOW" T)
     FIRSTCONF←'T;

IF FIRSTCONF = 'NIL
THEN MARK←'("Threat to C3 Nodes: VERY LOW" T);

THEN (PutValue BSSBoard 'COMMENT (APPEND (GetValue BSSBoard
 'COMMENT) MARK))    (SETQ HYPOTRANSFER 'NIL);

(* This section will map each event to the appropriate
  hypothesis for processing.)

IF (GETPROP EVENTMATCH 'P2HYPO) = 'HOSTILE
THEN (RunRS 'BSSHOSTILE);

IF (GETPROP EVENTMATCH 'P2HYPO) = 'TEST
THEN (RunRS 'BSSTEST);

IF (GETPROP EVENTMATCH 'P2HYPO) = 'UNRESOLVED
THEN (RunRS 'BSSUNRESOLVED);

IF (GETPROP EVENTMATCH 'P2HYPO) = 'UNKNOWN
THEN (RunRS 'BSSUNKNOWN);
#&(RuleSetSource "SWU0.zI[:.<;:.6")
63←
NIL
63←
NIL
63←
NIL
63←
NIL
63←
NIL
63←
NIL
63←
NIL
63←
NIL
63←
NIL
63←
NIL
63←
NIL
63←(REDO 100)
```

```
using $BSSUNKNOWN
RuleSet BSSUNKNOWN
Last Edited: 9-Oct-85 15:17:21 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   MARK READ OUTVAR XVAR YVAR;
Control Structure: DOALL;
********



(* BSSUNKNOWN processes the BSS message for those events wh
ich have been identified as UNKNOWN. First, BSSUNKNOWN does
 confidence level revisionand then consults BSSBoard quite
heavily to determine when hypothesis revision to other case
s such as HOSTILE, TEST or UNRESOLVED is in order. BSSUNRES
OLVED then fills out the BSS report for the MWO and permits
 explanation using Explain and Explain2.)

 THEN (RunRS 'BoardVoting);

IF HIGHVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

IF HIGHVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF LOWVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF LOWVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

(* Having accomplshed confidence level revision above, now
 belief revision occurs when warranted.)

IF BSSBoard:EXPNUMBER <= 2
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
   HYPOTRANSFER='NIL
THEN (SETQ HYPOTRANSFER 'T) (RunRS 'BSSTEST) (STOP);

IF BSSBoard:EXPNUMBER <= 2
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
   HYPOTRANSFER='NIL
THEN (SETQ HYPOTRANSFER 'T)(RunRS 'BSSTEST) (STOP);
```

```
THEN (SETQ HYPOTRANSFER 'T)(RunRS 'BSSTEST) (STOP);

IF HYPOTRANSFER = 'NIL
   BSSBoard:EXPNUMBER > 2
THEN (SETQ HYPOTRANSFER 'T) (RunRS 'BSSHOSTILE) (STOP);

IF HYPOTRANSFER = 'NIL
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN (SETQ HYPOTRANSFER 'T) (RunRS 'BSSUNRESOLVED) (STOP);

IF HYPOTRANSFER = 'NIL
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN (SETQ HYPOTRANSFER 'T) (RunRS 'BSSUNRESOLVED) (STOP);

(* Threat is assumed to be NO unless proved otherwise.)
THEN OUTVAR←'NO;

IF BSSBoard:EXPNUMBER > 2
THEN OUTVAR←'YES;

(* The C3 Targets and rest of the US targets are combined
 on the three BSSBoard lists.)

IF TRENDBoard:WILLNUMBER >= 1
THEN (PutValue BSSBoard 'WILLDETECT (APPEND BSSBoard:WILLDE
TECT TRENDBoard:WILLDETECT));

IF TRENDBoard:CLOSENUMBER >= 1
THEN (PutValue BSSBoard 'CLOSEDETECT (APPEND BSSBoard:CLOSE
DETECT  TRENDBoard:CLOSEDETECT));

IF TRENDBoard:FARNUMBER >= 1
THEN (PutValue BSSBoard 'FARDETECT (APPEND BSSBoard:FARDETE
CT TRENDBoard:FARDETECT));

(* Output Section.)
THEN (PRINTOUT T T "EVENT Considered: UNKNOWN" T T T"      B
SS  " EVENTMATCH T T "1. DETECTED: " P3TIME T "2. BSS: "  P
3SITENUM T "3. THREAT: " OUTVAR  T "4. LAUNCHER TYPE: " MAT
CHBoard:LAUNCHERTYPE T "5. OBJECTS TRACKED: " P3OBJNUM T)
MARK←'NIL;

IF BSSBoard:WILLDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. VERY LIKELY TARGETS: "(CONVERT
 BSSBoard:WILLDETECT) T);

IF MARK = 'NIL
   BSSBoard:CLOSEDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. LIKELY TARGETS: " (CONVERT BSS
Board:CLOSEDETECT) T);
```

```
Board:CLOSEDETECT) T);

IF MARK = 'NIL
    BSSBoard:FARDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. POSSIBLE TARGETS: " (CONVERT
BSSBoard:FARDETECT) T);

IF MARK = 'NIL
THEN (PRINTOUT T "6. POSSIBLE TARGETS: NONE" T);

THEN (PRINTOUT T "7. EST. IMPACT TIME: " P3IMPACTTIME T "8.
 CONFIDENCE: " (GETPROP EVENTMATCH 'P2CONFIDENCE) T T) (Run
RS 'Speculate);

IF EXPLAINBoard:COMMENT ~= 'NIL
THEN (EVAL EXPLAINBoard:COMMENT);

THEN(PRINTOUT T T "If you wish further explanation type T,"
 T "Otherwise type NIL to proceed." T)(SETQ READ (RATOM NIL
 NIL));

IF READ = 'T
THEN (SETQ FACTORLIST (LIST BSSBoard:EXPNUMBER 'BSSBoard  T
RENDBoard:EXPNUMBER 'TRENDBoard MATCHBoard:EXPNUMBER 'MATCH
Board IntelBoard:EXPNUMBER 'IntelBoard SYSBoard:EXPNUMBER '
SYSBoard));

IF READ = 'T
   OUTVAR = 'NO
THEN (RunRS 'LOWRanking) (RunRS 'Explain) (STOP);

IF READ = 'T
   OUTVAR = 'YES
THEN (RunRS 'HIGHRanking) (RunRS 'Explain);
#&(RuleSetSource (10 . 53880))
43←
NIL
43←
NIL
43←
NIL
43←
NIL
43←
NIL
43←
NIL
43←
NIL
43←(REDO 100)
```

using $BSSUNRESOLVED

RuleSet BSSUNRESOLVED

```
Last Edited: 9-Oct-85 14:10:37 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   MARK READ OUTVAR XVAR YVAR;
Control Structure: DOALL;
********


(* BSSUNRESOLVED processes the BSS message for those events
 which have been identified as UNRESOLVED. First, BSSUNRESO
LVED consults BoardVoting to determine when confidence leve
l revision is appropriate, and then consults BSSBoard quite
 heavily to determine when hypothesis revision to other cas
es such as HOSTILE, TEST or UNKNOWN is in order. BSSTEST th
en fills out the BSS report for the MWO and permits explana
tion using Explain and Explain2.)

 THEN (RunRS 'BoardVoting);

IF HIGHVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

IF HIGHVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF LOWVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF LOWVOTE >= 3
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

(* Having revised the confidence when necessary above,
  hypothesis revision takes place below.)

IF BSSBoard:EXPNUMBER <= 2
   (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
   HYPOTRANSFER = 'NIL
THEN (SETQ HYPOTRANSFER 'T)(RunRS 'BSSTEST) (STOP);

IF BSSBoard:EXPNUMBER <= 2
  (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
  HYPOTRANSFER = 'NIL
THEN (SETQ HYPOTRANSFER 'T) (RunRS 'BSSTEST) (STOP);
```

```
IF HYPOTRANSFER = 'NIL
   BSSBoard:EXPNUMBER > 2
THEN (SETQ HYPOTRANSFER 'T) (RunRS 'BSSHOSTILE) (STOP);

IF HYPOTRANSFER = 'NIL
   TRENDBoard:EXPNUMBER <= 2
THEN (SETQ HYPOTRANSFER 'T) (RunRS 'BSSUNKNOWN) (STOP);

(* The contents of BSSBoard can be consilidated with both
  the C3 Threat and threat to other targets.)

THEN OUTVAR←'NO;



IF BSSBoard:EXPNUMBER > 2
THEN OUTVAR←'YES;

IF TRENDBoard:WILLNUMBER >= 1
THEN (PutValue BSSBoard 'WILLDETECT (APPEND BSSBoard:WILLDE
TECT TRENDBoard:WILLDETECT));

IF TRENDBoard:CLOSENUMBER >= 1 THEN (PutValue BSSBoard 'CLO
SEDETECT (APPEND BSSBoard:CLOSEDETECT  TRENDBoard:CLOSEDETE
CT));

IF TRENDBoard:FARNUMBER >= 1 THEN (PutValue BSSBoard 'FARDE
TECT (APPEND BSSBoard:FARDETECT TRENDBoard:FARDETECT));

(* The output section shows output to MWO and explains.)

THEN (PRINTOUT T T "EVENT Considered: UNRESOLVED" T T T "
  BSS  " EVENTMATCH T T "1. DETECTED: " P3TIME T "2. BSS: "
  P3SITENUM T "3. THREAT: " OUTVAR  T "4. LAUNCHER TYPE: "
MATCHBoard:LAUNCHERTYPE T "5. OBJECTS TRACKED: " P3OBJNUM T
)  MARK←'NIL;

IF BSSBoard:WILLDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. VERY LIKELY TARGETS: "(CONVERT
 BSSBoard:WILLDETECT) T);

IF MARK = 'NIL
   BSSBoard:CLOSEDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. LIKELY TARGETS: " (CONVERT BSS
Board:CLOSEDETECT) T);

IF MARK = 'NIL
    BSSBoard:FARDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. POSSIBLE TARGETS: " (CONVERT
```

```
IF MARK = 'NIL
    BSSBoard:FARDETECT ~= 'NIL
THEN MARK←'T (PRINTOUT T "6. POSSIBLE TARGETS: " (CONVERT
BSSBoard:FARDETECT) T);

IF MARK = 'NIL
THEN (PRINTOUT T "6. POSSIBLE TARGETS: NONE" T);

THEN (PRINTOUT T "7. EST. IMPACT TIME: " P3IMPACTTIME T "8.
 CONFIDENCE: " (GETPROP EVENTMATCH 'P2CONFIDENCE) T T) (Run
RS 'Speculate);

(* What about a Sprculate2 rule set for the above.)

IF EXPLAINBoard:COMMENT ~= 'NIL
THEN (EVAL EXPLAINBoard:COMMENT);

THEN (PRINTOUT T T "If you wish further explanation type T,
" T "Otherwise type NIL to proceed." T) (SETQ READ (RATOM N
IL NIL));

IF READ = 'T
THEN (SETQ FACTORLIST (LIST BSSBoard:EXPNUMBER 'BSSBoard  T
RENDBoard:EXPNUMBER 'TRENDBoard MATCHBoard:EXPNUMBER 'MATCH
Board IntelBoard:EXPNUMBER 'IntelBoard SYSBoard:EXPNUMBER '
SYSBoard));

IF READ = 'T
   OUTVAR = 'NO
THEN (RunRS 'LOWRanking) (RunRS 'Explain) (STOP);

IF READ = 'T
   OUTVAR = 'YES
THEN (RunRS 'HIGHRanking) (RunRS 'Explain);
#&(RuleSetSource (10 . 55452))
73←
NIL
73←
NIL
73←
NIL
73←
NIL
73←
NIL
73←
NIL
73←
NIL
73←(REDO 100)
```

```
17←(REDO 16)
Using $C3Analyst
RuleSet C3Analyst
Last Edited: 3-Oct-85 18:07:46 by
WorkSpace Class: Reports;
Compiler Options:  ;
Temporary Vars:   A1LIST A2LIST A3LIST A4LIST B1LIST B2LIST
 B3LIST B4LIST STATUS METALIST;
Control Structure: DOALL;
********



(* C3Analyst is invoked whenever an SYS report about a C3 n
ode is received. C3Analyst puts together the lists that com
prise the 8 C3 paths which are then handed to SYSCheck to d
etermine what their status is, which is maintained by C3Boa
rd. At the completion, SYSBoard is filled with the C3 netwo
rk variables and the COMMENT ammended.)

THEN (SETQ B1LIST '(TGT3 TGT4 TGT6))
     (SETQ A1LIST '(TGT3 TGT4 TGT5))
     (SETQ B2LIST '(TGT3 TGT4 TGT2 TGT6))
     (SETQ A2LIST '(TGT3 TGT4 TGT2 TGT5))
     (SETQ B3LIST '(TGT3 TGT2 TGT6))
     (SETQ A3LIST '(TGT3 TGT2 TGT5))
     (SETQ B4LIST '(TGT3 TGT2 TGT4 TGT6))
     (SETQ A4LIST '(TGT3 TGT2 TGT4 TGT5))
     (SETQ METALIST '(A1LIST B1LIST A2LIST B2LIST A3LIST B3
LIST A4LIST B4LIST))
     SYSBoard:AZERO←0   SYSBoard:BZERO←0
     SYSBoard:APARTIAL←0  SYSBoard:BPARTIAL←0
     SYSBoard:AFULL←0   SYSBoard:BFULL←0;

THEN (SETQ STATUS 'FULL)
     (RunRS 'SYSCheck)
    SYSBoard:AGSNUMBER←1 SYSBoard:BGSNUMBER←1;

 (* Default Condition above set AGSNUMBER and BGSNUMBER
    to one. It is modified in the six rules below.)

IF SYSBoard:AFULL >= 1
THEN SYSBoard:AGSNUMBER←4;

IF SYSBoard:BFULL >= 1
THEN SYSBoard:BGSNUMBER←4;

IF SYSBoard:APARTIAL >= 2
   SYSBoard:AFULL = 0
THEN SYSBoard:AGSNUMBER←3;
```

```
   SYSBoard:AFULL = 0
THEN SYSBoard:AGSNUMBER←3;

IF SYSBoard:BPARTIAL >= 2
   SYSBoard:BFULL = 0
THEN SYSBoard:BGSNUMBER←3;

IF SYSBoard:APARTIAL = 1
   SYSBoard:AFULL = 0
THEN SYSBoard:AGSNUMBER←2;

IF SYSBoard:BPARTIAL = 1
   SYSBoard:BFULL = 0
THEN SYSBoard:BGSNUMBER←2;

THEN (PRINTOUT T T "this is AGSnumber: " SYSBoard:AGSNUMBER
 T "This is BGSnumber: " SYSBoard:BGSNUMBER T)
  (SETQ FNUMBER SYSBoard:BGSNUMBER)
  (SETQ SNUMBER SYSBoard:AGSNUMBER)
  (RunRS 'Combinatorial);

IF SUMNUMBER = 4
THEN (SETQ C3COMMENT '("C3 Network Capability: FULL" T))(Pu
tValue SYSBoard 'COMMENT (APPEND (GetValue SYSBoard 'COMMEN
T) C3COMMENT));

IF SUMNUMBER = 3
THEN (SETQ C3COMMENT '("C3 Network Capability: PARTIAL"T))
(PutValue SYSBoard 'COMMENT (APPEND (GetValue SYSBoard 'COM
MENT) C3COMMENT));

IF SUMNUMBER = 2
THEN (SETQ C3COMMENT '("C3 Network Capability: CRIPPLED" T)
) (PutValue SYSBoard 'COMMENT (APPEND (GetValue SYSBoard 'C
OMMENT) C3COMMENT));

IF SUMNUMBER = 1
THEN (SETQ C3COMMENT '("C3 Network Capability: OUT" T))
(PutValue SYSBoard 'COMMENT (APPEND (GetValue SYSBoard 'COM
MENT) C3COMMENT));
#&(RuleSetSource "GWU0.zI[:.PG8.29")
21←
NIL
21←
NIL
21←
NIL
21←
NIL
21←(REDO 97)
```

using $C3Board

RuleSet C3Board

Last Edited: 3-Oct-85 18:09:00 by

```
WorkSpace Class: Reports;
Compiler Options:  ;
Temporary Vars:  ;
Control Structure: DOALL;
********


(* C3BOARD updates the number of C3 path lists which falls
into the categories of FULL, PARTIAL or ZERO for both AGS a
nd BGS path links.)

IF TAG = 'BGS
   STATUS = 'FULL
THEN SYSBoard:BFULL←(PLUS SYSBoard:BFULL 1);

IF TAG = 'BGS
   STATUS = 'PARTIAL
THEN SYSBoard:BPARTIAL←(PLUS SYSBoard:BPARTIAL 1);

IF TAG = 'BGS
   STATUS = 'ZERO
THEN SYSBoard:BZERO←(PLUS SYSBoard:BZERO 1);

IF TAG = 'AGS
   STATUS = 'FULL
THEN SYSBoard:AFULL←(PLUS SYSBoard:AFULL 1);

IF TAG = 'AGS
   STATUS = 'PARTIAL
THEN SYSBoard:APARTIAL←(PLUS SYSBoard:APARTIAL 1);

IF TAG = 'AGS
   STATUS = 'ZERO
THEN SYSBoard:AZERO←(PLUS SYSBoard:AZERO 1);
#&(RuleSetSource "GWU0.zI[:.PG8.30")
23←
NIL
23←
NIL
23←
NIL
23←
NIL
23←
NIL
23←(REDO 97)
```

using $Combinatorial

RuleSet Combinatorial

Last Edited: 3-Oct-85 18:10:35 by

```
WorkSpace Class: Sensors;
Compiler Options:  ;
Temporary Vars:  ;
Control Structure: DO1;
********


(* This rule set takes two facts, a more important fact kno
wn as FNUMBER that is abstracted to 4 states, and a less im
portant fact, SNUMBER also known in 4 discrete states, that
 aret then combined to form a discrete state known as SUMNU
MBER that is at a higher level of abstraction.)

IF FNUMBER = 4
   SNUMBER > 2
THEN (SETQ SUMNUMBER 4);

IF FNUMBER = 4
THEN (SETQ SUMNUMBER 3);

IF FNUMBER = 3
   SNUMBER > 2
THEN (SETQ SUMNUMBER 3);

IF FNUMBER = 3
THEN (SETQ SUMNUMBER 2);

IF FNUMBER = 2
   SNUMBER = 4
THEN (SETQ SUMNUMBER 3);

IF FNUMBER = 2
   SNUMBER > 1
THEN (SETQ SUMNUMBER 2);

IF FNUMBER = 2
THEN (SETQ SUMNUMBER 1);

IF FNUMBER = 1
   SNUMBER > 2
THEN (SETQ SUMNUMBER 2);

IF FNUMBER = 1
THEN (SETQ SUMNUMBER 1);
#&(RuleSetSource "GWU0.zI[:.PG8.31")
27←
```

```
29←(PP CONVERT)

(CONVERT
  [LAMBDA (C3LIST)                                   (* edited: " 3-Sep-85 14:02")
    (PROG (MATCHLIST MATCHPOINT MATCHOUT MATCHVAR)

              (* CONVERT takes a list or a single item expressed in the property t
ag format of SENTINEL such as TGT3, LS5 or TGT04 and converts them to a list o
r object that is composed of the name of the object such as: BRF KAPUSTINYAR B
SS4 etc.)


          (SETQ MATCHLIST C3LIST)
          (COND
            ((NOT (LISTP MATCHLIST))
              (SETQ MATCHOUT (GETPROP MATCHLIST
                                      (QUOTE NAME)))
              (RETURN MATCHOUT)))
          (COND
            ((NULL (CDR MATCHLIST))
              (SETQ MATCHOUT (GETPROP (CAR MATCHLIST)
                                      (QUOTE NAME)))
              (SETQ MATCHLIST NIL)))
      LOOP(COND
            ((NULL (CAR MATCHLIST))
              (RETURN MATCHOUT)))
          (SETQ MATCHPOINT (CAR MATCHLIST))
          (SETQ MATCHVAR (GETPROP MATCHPOINT (QUOTE NAME)))
          (SETQ MATCHOUT (APPEND (LIST MATCHVAR)
                                 MATCHOUT))
          (SETQ MATCHLIST (CDR MATCHLIST))
          (GO LOOP])
(CONVERT)
30←
NIL
30←
NIL
30←
NIL
30←
NIL
30←
NIL
30←
NIL
30←
NIL
30←
NIL
30←(REDO 97)
```

using $Countries

RuleSet Countries

Last Edited: 3-Oct-85 18:03:53 by

```
WorkSpace Class: Targets;
Compiler Options:  ;
Temporary Vars:  ;
Control Structure: D01;
********


(* This ruleset takes the country number reported from the
INTEL reports and converts it to a country name)

IF COUNTRYNUM = 17
THEN (SETQ COUNTRY 'THEM);

IF COUNTRYNUM = 19
THEN (SETQ COUNTRY 'US);

IF COUNTRYNUM = 0
THEN (SETQ COUNTRY 'UNKNOWN);

IF COUNTRYNUM = 1
THEN (SETQ COUNTRY 'ANYBODY);

IF COUNTRYNUM = 2
THEN (SETQ COUNTRY 'ANYONE);

IF COUNTRYNUM = 3
THEN (SETQ COUNTRY 'ANYPLACE);

IF COUNTRYNUM = 4
THEN (SETQ COUNTRY 'BACKLAND);

IF COUNTRYNUM = 5
THEN (SETQ COUNTRY 'BOTTOMLAND);

IF COUNTRYNUM = 6
THEN (SETQ COUNTRY 'EASTLAND);

IF COUNTRYNUM = 7
THEN (SETQ COUNTRY 'INLAND);

IF COUNTRYNUM = 8
THEN (SETQ COUNTRY 'NOBODY);

IF COUNTRYNUM = 9
THEN (SETQ COUNTRY 'NOPLACE);
```

```
IF COUNTRYNUM = 8
THEN (SETQ COUNTRY 'NOBODY);

IF COUNTRYNUM = 9
THEN (SETQ COUNTRY 'NOPLACE);

IF COUNTRYNUM = 10
THEN (SETQ COUNTRY 'NORTHLAND);

IF COUNTRYNUM = 11
THEN (SETQ COUNTRY 'OUTBACK);

IF COUNTRYNUM = 12
THEN (SETQ COUNTRY 'OVERTHERE);

IF COUNTRYNUM = 13
THEN (SETQ COUNTRY 'SOMEBODY);

IF COUNTRYNUM = 14
THEN (SETQ COUNTRY 'SOMEONE);

IF COUNTRYNUM = 15
THEN (SETQ COUNTRY 'SOMEPLACE);

IF COUNTRYNUM = 16
THEN (SETQ COUNTRY 'SOUTHLAND);

IF COUNTRYNUM = 18
THEN (SETQ COUNTRY 'TOPLAND) ;

IF COUNTRYNUM = 20
THEN (SETQ COUNTRY 'WESTLAND) ;
#&(RuleSetSource "LVU0.zI[:.P9B.28")
36←
NIL
36←
NIL
36←
NIL
36←
NIL
36←
NIL
36←
NIL
36←
NIL
36←
NIL
36←(REDO 97)
```

```
using $DetectCheck
```

RuleSet DetectCheck

```
Last Edited: 3-Oct-85 18:40:55 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   TAGLIST HEADING MARK HIVALCOMMENT;
Control Structure: DOALL;
********


(* DetectCheck processes the BSSTAG and the HIVALTAG of the
 launchsite by calling on HeadingCheck and TargetCheck resp
ectively. The values obtained are posted on the DetectBoard
 for the BSS sites and on the TRENDBoard for High Value Tar
gets.)

THEN TAGLIST←(GETPROP MATCHBoard:P2LS 'BSSTAG)
     HEADING←MATCHBoard:LSHEADING
     DetectBoard:EXPNUMBER←1;

IF HEADING = 'XXX
THEN DetectBoard:EXPNUMBER←0
     TRENDBoard:EXPNUMBER←0
     DetectBoard:COMMENT←'(PRINTOUT T "NO HEADING Informati
on available yet." T "Cannot predict BSS detection possibil
ities." T)(STOP);

THEN  (RunRS 'HeadingCheck) MARK←'T;

(* After running HeadingCheck various properties such as WI
LLDETECT, CLOSEDETECT and FARDETECT are analyzed.)
IF DetectBoard:WILLDETECT ~= 'NIL
THEN DetectBoard:EXPNUMBER←4
 (SETQ DETECTCOMM (CONVERT (GetValue DetectBoard 'WILLDETEC
T)))   DetectBoard:COMMENT←'(PRINTOUT T "BSS Detection Oppo
rtunities are: VERY STRONG" T "Occuring at BSS sites: " DET
ECTCOMM T);

 IF DetectBoard:WILLDETECT = 'NIL DetectBoard:CLOSEDETECT ~
= 'NIL THEN DetectBoard:EXPNUMBER←3(SETQ DETECTCOMM (CONVER
T (GetValue DetectBoard 'CLOSEDETECT)))DetectBoard:COMMENT←
'(PRINTOUT T "BSS Detection Opportunities are: STRONG" T "O
curring at BSS sites: " DETECTCOMM T);

 IF DetectBoard:WILLDETECT = 'NIL
 DetectBoard:CLOSEDETECT = 'NIL
 DetectBoard:FARDETECT ~= 'NIL
 THEN DetectBoard:EXPNUMBER←2
 (SETQ DETECTCOMM (CONVERT (GetValue DetectBoard 'FARDETECT
           DetectBoard:COMMENT←'(PRINTOUT T "BSS Detection O
  pportunities are: POOR" T "Coverage of these BSS sites devi
  ates from the reported heading by up to 10 degrees: " DETEC
  TCOMM T);
```

```
|);

 IF DetectBoard:EXPNUMBER = 1
THEN  (SETQ DETECTCOMM(CONVERT (GetValue DetectBoard 'NEARP
T)))
 DetectBoard:COMMENT←'(PRINTOUT T "BSS Detection Opportunit
es are: VERY POOR" T "The closest BSS site coverage is at:
" DETECTCOMM  T);

 (* Special case when DETECTPT is empty it is set to 0)
IF DetectBoard:DETECTPT = 'L THEN DetectBoard:DETECTPT←0;

(* TargetCheck is run to determine what C3 targets fall int
o the category of WILLDETECT, CLOSEDETECT and FARDETECT. Re
sults are interpreted below.)
 THEN TAGLIST←(GETPROP MATCHBoard:P2LS 'HIVALTAG)
     (RunRS 'TargetCheck);

 IF TRENDBoard:WILLNUMBER >= 1
 THEN TRENDBoard:HIVALNUMBER←4 MARK←'NIL(SETQ HIVALCOMMENT
'("Threat to C3 Nodes: VERY HIGH" T))(PutValue DetectBoard
'COMMENT (APPEND (GetValue DetectBoard 'COMMENT) HIVALCOMME
NT));

IF TRENDBoard:CLOSENUMBER >= 1
 MARK = 'T
THEN TRENDBoard:HIVALNUMBER←3 MARK←'NIL (SETQ HIVALCOMMENT
'("Threat to C3 Nodes: HIGH" T))(PutValue DetectBoard 'COMM
ENT (APPEND (GetValue DetectBoard 'COMMENT) HIVALCOMMENT));

IF TRENDBoard:FARNUMBER >= 1
 MARK = 'T
 THEN TRENDBoard:HIVALNUMBER←2 MARK←'NIL(SETQ HIVALCOMMENT
'("Threat to C3 Nodes: LOW" T))(PutValue DetectBoard 'COMME
NT (APPEND (GetValue DetectBoard 'COMMENT)  HIVALCOMMENT));

IF MARK = 'T
 THEN TRENDBoard:HIVALNUMBER←1 (SETQ HIVALCOMMENT '("Threat
 to C3 Nodes: VERY LOW" T))(PutValue DetectBoard 'COMMENT (
APPEND (GetValue DetectBoard 'COMMENT) HIVALCOMMENT));

THEN (PRINTOUT T "These sites will detect: " (CONVERT Detec
tBoard:WILLDETECT) T "These site could detect: " (CONVERT D
etectBoard:CLOSEDETECT) T "These site will probably not det
ect: " (CONVERT DetectBoard:FARDETECT) T "The BSS Number fo
r ADS2 is: " DetectBoard:DETECTPT T "The near point is: " (
CONVERT DetectBoard:NEARPT) T "with a diference of " Detect
Board:NEARDIFF T "degrees from the reported HEADING." T);

THEN (EVAL (GetValue DetectBoard 'COMMENT));
```

```
NIL
41←(REDO 37)

(EVENTCHECK
  [LAMBDA NIL                                    (* edited: "10-Oct-85 17:18")
    (PROG (MATCHLIST MATCHPOINT MATCHTIME BSSTIME)

          (* When no ADS2 report can be matched to a previous ADS1 report;
 EVENTCHECK lets the user know what events are there in the EVENTLIST, which
 ones are not yet matched, and which ones had satellite coverage incompatibility
 with the reported launchsite. TIMECHECK is also invoked to delete any events from
 the active list whose ADS2 reports are overdue. Finally, EVENTCHECK permits the
 user to match any previously unmatched report. This should be used mainly in the
 training mode.)


          (COND
            ((NULL EVENTLIST)
              (RETURN (printout T T
"No event reported since SENTINEL DECISION AID
was initialized."
                                           T)))
            (T (printout T T
"These are the past EVENTS since SENTINEL was initialized:
"
                                    EVENTLIST T T)))
          (SETQ MATCHLIST EVENTLIST)
          (SETQ TESTLIST NIL)

          (* TESTLIST will contain all the ADS1 events which have not yet be
en matched to an ADS2 report. The loop checks all of the events one at a time to
see if they have been matched. For matched events P1TIME of the EVENT propert
y list is NIL)


      LOOP[COND
            ((NULL (CAR MATCHLIST))
              (COND
                ((NULL TESTLIST)
                  (RETURN (printout T
                "All events have already been matched."
                                            T
        "SENTINEL concludes this is a spurious event."
                                            T
                      "No further processing underway."
                                            T T)))
                (T (RETURN (printout T T
"These are the events which have not been matched to an ADS
2 Report."
                                             T T TESTLIST T]
          (SETQ MATCHPOINT (CAR MATCHLIST))
          (SETQ MATCHTIME (GETPROP MATCHPOINT
```

```
          (SETQ BSSTIME (GETPROP MATCHPOINT (QUOTE P1TIME))
            )
          [COND
            ((NULL MATCHTIME)
              (COND
                ((NOT (NULL BSSTIME))
                  (SETQ TESTLIST (APPEND TESTLIST
                                         (LIST MATCHPOINT]
          (SETQ MATCHLIST (CDR MATCHLIST))
          (GO LOOP))
```

(* This procedure checks on satellite compatibility; i,e, if ADS2 report came from the ADSNORTH satellite, the LATITUDE of the matched launch site should be between 15 and 28,5 degrees; similarly for the ADSSOUTH, the LATITUDE of the matched report should be between 27,5 and 40 degrees, Note that there is a small area of overlapping coverage around 28 degrees,)

```
    (PROG (P1LS LATLNCH MATCHPOINT MATCHLIST FLAGLIST)
          (SETQ MATCHLIST TESTLIST)
          (SETQ FLAGLIST NIL)
      LOOP[COND
            ((NULL (CAR MATCHLIST))
              (COND
                ((NULL FLAGLIST)
                  (RETURN))
                NIL
                (T (RETURN (printout T T
"These events were not matched because the reported
LATITUDE of their launchsites did not match the
NORMAL ADS2 satellite coverage: "
                                         T FLAGLIST T]
          (SETQ MATCHPOINT (CAR MATCHLIST))
          (SETQ P1LS (GETPROP MATCHPOINT (QUOTE P1LS)))
          (SETQ LATLNCH (GETPROP P1LS (QUOTE LAT)))
```

(* If all the events are matched; TESTLIST is NIL; but it still crunches through this code, Therefore, LATLNCH must be set to a number so that the GREATERP and LESSP functions can operate)

```
          (COND
            ((NULL LATLNCH)
              (SETQ LATLNCH 28)))
          [COND
            ((EQ (GNC SENSORNUMBER)
                 (QUOTE S))
              (SETQ SENSORCAP (GETPROP (QUOTE ADSSOUTH)
                                       (QUOTE CAPABILITY)))
              (SETQ REASONOUT (GETPROP (QUOTE ADSSOUTH)
                                       (QUOTE REASON)))
```

```
                (COND
                  ((LESSP LATLNCH (QUOTE 27.5))
                    (SETQ FLAGLIST (APPEND FLAGLIST
                                           (LIST MATCHPOINT]
          [COND
            ((EQ (GNC SENSORNUMBER)
                 (QUOTE N))
              (SETQ SENSORCAP (GETPROP (QUOTE ADSNORTH)
                                       (QUOTE CAPABILITY)))
              (SETQ REASONOUT (GETPROP (QUOTE ADSNORTH)
                                       (QUOTE REASON)))
              (COND
                ((GREATERP LATLNCH (QUOTE 29.5))
                  (SETQ FLAGLIST (APPEND FLAGLIST
                                         (LIST MATCHPOINT]
          (SETQ MATCHLIST (CDR MATCHLIST))
          (GO LOOP))
    (TIMECHECK)                              (* TIMECHECK is invoked; and th
e user is permitted to match the ADS2 Report to any unmatched (ADS1) EVENT.)
    (COND
      ((NULL TESTLIST)
        (RETURN))
      (T (printout T T "Any of the unmatched events: " T
                   TESTLIST T T
   "can now be matched with the received ADS2 Report."
                   T
"Type in NIL if you do not want to match this report."
                   T
"Or, type in the complete EVENTnumber of the EVENT you wish
 to match it with."
                   T
   "DO NOT put parantheses around your response ! ! !"
                   T)
         (SETQ READ (RATOM NIL NIL))
                                             (* With a NIL, the user exits out
of the program)
         (COND
           ((NULL READ)
             (printout T "EVENT returned unmatched." T))
           (T (COND
                ([NOT (NULL (GETPROP READ (QUOTE P1TIME]
                  (SETQ EVENTMATCH READ)

         (* MATCH is made by removing P1TIME from the EVENT property li
st and then running the ADS2Confidence rule set; before this is done the prelimina
ry steps normally done by ADS2HANDLER are duplicated here.)


                  (REMPROP EVENTMATCH (QUOTE P1TIME))
                  (PUTPROP EVENTMATCH (QUOTE P2TIME)
                           P2TIME)
```

```
    ((NULL READ)
      (printout T "EVENT returned unmatched." T))
    (T (COND
         ([NOT (NULL (GETPROP READ (QUOTE P1TIME]
           (SETQ EVENTMATCH READ)
```

(* MATCH is made by removing P1TIME from the EVENT property list and then running the ADS2Confidence rule set; before this is done the preliminary steps normally done by ADS2HANDLER are duplicated here.)

```
           (REMPROP EVENTMATCH (QUOTE P1TIME))
           (PUTPROP EVENTMATCH (QUOTE P2TIME)
                    P2TIME)
           (PutValue MATCHBoard (QUOTE LAUNCHSITE)
                     (GETPROP EVENTMATCH
                              (QUOTE P1LS)))
           (PutValue MATCHBoard (QUOTE P2LS)
                     (GETPROP EVENTMATCH
                              (QUOTE P1LS)))
           (PutValue MATCHBoard (QUOTE RPTSENSORCAP)
                     SENSORCAP)
           (PutValue MATCHBoard (QUOTE OUTAGEREASON)
                     REASONOUT)
           (PutValue MATCHBoard (QUOTE LSHEADING)
                     P2HEADING)
           (PutValue MATCHBoard (QUOTE LSBSS)
                     P2BSS)
           (RunRS (QUOTE SensorCheck))
           (RunRS (QUOTE DetectCheck))
           (RunRS (QUOTE ADS2Confidence)))
         (T (printout T
                 "INCORRECT EVENTNUMBER typed in."
                         T "what was read in is: "
                         READ T T
    "Response MUST be a NIL or the EVENTnumber,"
                         T
                 "In this form: EVENT3 (example) "
                         T "Do not use parantheses " T
                       "EVENTCHECK invoked again."
                         T)     (* IF incorrect response is typed in, the user is made aware of the situation and this procedure is invoked again.)
            (EVENTCHECK])
(EVENTCHECK)
52←
NIL
52←
NIL
52←
NIL
52←(REDO 97)
```

```
using $Explain
```

RuleSet Explain

Last Edited: 3-Oct-85 18:27:06 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   MARK READ OUTVAR;
Control Structure: DOALL;
********


(* Explain permits the MWO to access 2 layers of explanatio
n when answering ADS-2 reports. Explain contains the higher
, abstract level of explanation which can be foundin the CO
MMENT of each blackboard. Explain calls on      Explain2 to
 provide the second, more detailed layer.    WHY questions
answered by Explain, HOW questions answered by Explain2. A
Nil response exits the program.)

IF EXPLAINBoard:COMMENT ~= 'NIL
THEN (EVAL EXPLAINBoard:COMMENT);

THEN (PRINTOUT T T "Explanation for ADS-2 processing is in
two layers." T "The first, abstract layer can be accessed b
y typing in" T "WHY followed by a carriage return. <CR>" T
"The second detailed layer can be accessed from" T "the fir
st layer by typing in HOW <CR>." T "Typing NIL <CR> exits t
he program." T T   "Type in  WHY or NIL now: " T)
   MARK←'NIL
   (SETQ READ (RATOM NIL NIL));

IF READ = 'NIL
THEN (STOP);

IF READ = 'WHY
THEN OUTVAR←EXPLAINBoard:FACTOR1   MARK←'T
     (EVAL (GetValue (EVAL OUTVAR) 'COMMENT));

IF MARK = 'NIL
THEN (PRINTOUT T "INCORRECT RESPONSE RECEIVED." T)
     (RunRS 'Explain);

THEN (PRINTOUT T T "Type in WHY for more top layer explanat
ion" T "or HOW to see details of comment above." T "NIL is
exit: Type in response: " T) MARK←'NIL
      (SETQ READ (RATOM NIL NIL));

IF READ = 'HOW
THEN (SETQ FACTOR EXPLAINBoard:FACTOR1)
     (RunRS 'Explain2)
```

```
IF READ = 'HOW
THEN (SETQ FACTOR EXPLAINBoard:FACTOR1)
     (RunRS 'Explain2)
(PRINTOUT T T "Type WHY to continue inquiry: NIL to exit" T
)    (SETQ READ (RATOM NIL NIL));

IF READ = 'NIL
THEN (STOP);

IF READ = 'WHY
THEN OUTVAR←EXPLAINBoard:FACTOR2 MARK←'T
     (EVAL (GetValue (EVAL OUTVAR) 'COMMENT));

IF MARK = 'NIL
THEN (PRINTOUT T "INCORRECT RESPONSE RECEIVED." T)
     (RunRS 'Explain);

THEN (PRINTOUT T T "WHY to continue inquiry: HOW to see det
ails" T "NIL to exit"  T)
     MARK←'NIL
     (SETQ READ (RATOM NIL NIL));

IF READ = 'NIL
THEN (STOP);

IF READ = 'HOW
THEN (SETQ FACTOR EXPLAINBoard:FACTOR2)
     (RunRS 'Explain2)
(PRINTOUT T T "Type WHY to continue inquiry: NIL to exit" T
)   (SETQ READ (RATOM NIL NIL));

IF READ = 'NIL
THEN (STOP);

IF READ = 'WHY
THEN OUTVAR←EXPLAINBoard:FACTOR3 MARK←'T
     (EVAL (GetValue (EVAL OUTVAR) 'COMMENT));

IF MARK = 'NIL
THEN (PRINTOUT T "INCORRECT RESPONSE RECIEVED." T)
     (STOP);

(* Beginning here, we have to be concerned with the fact th
at sometimes FACTOR4 and FACTOR5 may be NIL. In that  case,
 we STOP Explain.)

THEN (PRINTOUT T T "Type WHY to continue inquiry: HOW to se
e Details." T)  MARK←'NIL
     (SETQ READ (RATOM NIL NIL))
     OUTVAR←EXPLAINBoard:FACTOR4;
```

```
IF READ = 'NIL
THEN (STOP);

IF READ = 'WHY
   OUTVAR = 'NIL
THEN (PRINTOUT T T "No more information available." T)
     (STOP);

IF READ = 'WHY
THEN MARK←'T
    (EVAL (GetValue (EVAL OUTVAR) 'COMMENT));

IF MARK = 'NIL
THEN (STOP);

THEN (PRINTOUT T T "Type WHY to continue inquiry: HOW to se
e Details." T) MARK←'NIL
     OUTVAR←EXPLAINBoard:FACTOR5
     (SETQ READ (RATOM NIL NIL));

IF READ = 'HOW
THEN (SETQ FACTOR EXPLAINBoard:FACTOR4)
     (RunRS 'Explain2)
(PRINTOUT T T "Type WHY to continue inquiry: NIL to exit: "
 T) (SETQ READ (RATOM NIL NIL));

IF READ = 'WHY
   OUTVAR = 'NIL
THEN (PRINTOUT T T "No more information available." T)
     (STOP);

IF READ = 'WHY
THEN (EVAL (GetValue (EVAL OUTVAR) 'COMMENT))
     MARK ←'T
  (PRINTOUT T T "All factors explained: type HOW for detail
s" T "or NIL to exit" T)
    (SETQ READ (RATOM NIL NIL));

IF MARK = 'NIL
THEN (STOP);

IF READ = 'HOW
THEN (SETQ FACTOR EXPLAINBoard:FACTOR5)
     (RunRS 'Explain2)
     (PRINTOUT T T "All explanation capability has been sho
wn." T);
#&(RuleSetSource "SIU0.zI[:.Je8.11")
61←
NIL
61←
```

using $Explain2

RuleSet Explain2

Last Edited: 7-Oct-85 14:51:55 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   MARK OUTVAR;
Control Structure: DOALL;
********


(* Explain Rule Set provides the difference condifence leve
l processors to invoke the second, more detailed layer of e
xplanation. By setting FACTOR to the appropate Blackboard,
 Explain will show the results of analyzed data that is upd
ated and maintained in all the blackboards.)

IF FACTOR = 'SYSBoard
THEN (PRINTOUT T "C3 Network Analysis shows:" T T "Status o
f the 4 paths to the ADSGSF" T "FULL STATUS: " (GetValue SY
SBoard 'AFULL) T "PARTIAL STATUS: " (GetValue SYSBoard 'APA
RTIAL) T "ZERO STATUS: " (GetValue SYSBoard 'AZERO) T T "St
atus of the 4 paths to the BSSGSF" T "FULL STATUS: "  (GetV
alue SYSBoard 'BFULL) T "PARTIAL STATUS: " (GetValue SYSBoa
rd 'BPARTIAL) T "ZERO STATUS: " (GetValue SYSBoard 'BZERO)
T)
(PRINTOUT T (GetValue SYSBoard 'OUTSENSOR) " out of 9 senso
r sites are OUT" T "with ZERO CAPABILITY." T T (GetValue SY
SBoard 'AFFSENSOR) " out of 9 sensor sites are AFFECTED" T
"with PARTIAL CAPABILITY." T T (GetValue SYSBoard 'OUTTOTAL
) " SYS reports have been acknowledged." T "Out of these re
ports:" T T (GetValue SYSBoard 'OUTEMERG) " have been EMERG
ENCY reports." T (GetValue SYSBoard 'OUTSABOTAGE) " have be
en due to SABOTAGE." T (GetValue SYSBoard 'OUTTEST) " have
been due to TESTING." T (GetValue SYSBoard 'OUTFAILURE) " h
ave been due to FAILURE." T);

IF FACTOR = 'IntelBoard
THEN (PRINTOUT T "Intelligence reports:" T T "Existence of
Condition: " (GetValue IntelBoard 'INTELCONDITION) T "as pr
ovided by " (GetValue IntelBoard 'SOURCE) " sources." T T "
This forecast is intended for use until: " (GetValue IntelB
oard 'INTELDAYTIME) " hours." T);

IF FACTOR = 'DetectBoard
   DetectBoard:WILLDETECT ~= 'NIL
THEN (PRINTOUT T "Reported heading falls into the range of
coverage" T "of BSS sites: " (CONVERT DetectBoard:WILLDETEC
T) T "BSS Detection will occur.  "  T "The site in best pos
ition to detect this launch is: " DetectBoard:DETECTPT T);
```

```
IF FACTOR = 'DetectBoard
   DetectBoard:WILLDETECT = 'NIL
   DetectBoard:CLOSEDETECT ~= 'NIL
THEN (PRINTOUT T "Coverage of these BSS sites are within" T
 "5 degrees of the reported heading: " (CONVERT DetectBoard
:CLOSEDETECT) T "BSS Detection could easily occur."  T "The
 site in best position to detect this launch isnumber: " De
tectBoard:DETECTPT T);

IF FACTOR = 'DetectBoard
   DetectBoard:FARDETECT ~= 'NIL
   DetectBoard:WILLDETECT = 'NIL
THEN (PRINTOUT T "Coverage of these BSS sites deviates" T "
from the reported heading by up to 10 degrees: " (CONVERT D
etectBoard:FARDETECT) T  "BSS Detection may be possible. "
T);

IF FACTOR = 'DetectBoard
   DetectBoard:FARDETECT = 'NIL
   DetectBoard:WILLDETECT = 'NIL
   DetectBoard:CLOSEDETECT = 'NIL
THEN (PRINTOUT T "BSS Detection Unlikely to occur." T "Clos
est approach to BSS site detection is at: site" (CONVERT De
tectBoard:NEARPT)  T "with a difference of " DetectBoard:NE
ARDIFF " degrees." T);

IF FACTOR = 'DetectBoard
   DetectBoard:DETECTPT ~= 0
   DetectBoard:STATUSDETECTPT ~= 'FULL
THEN (PRINTOUT T "However, the Current CAPABILITY of"
T "BSS site number " DetectBoard:DETECTPT " is: " DetectBoa
rd:STATUSDETECTPT T);

IF FACTOR = 'DetectBoard
   TRENDBoard:WILLDETECT ~= 'NIL
THEN (PRINTOUT T T "The following C3 sites lie within 1" T
"degree of the reported heading: " (CONVERT TRENDBoard:WILL
DETECT) T "Threat to these sites is: ALMOST CERTAIN." T);

IF FACTOR = 'DetectBoard
   TRENDBoard:CLOSEDETECT ~= 'NIL
THEN (PRINTOUT T T "These C3 sites lie within 5 degrees" T
"of the reported heading: " (CONVERT TRENDBoard:CLOSEDETECT
) T "Threat to these sites is: LIKELY." T);

IF FACTOR = 'DetectBoard
   TRENDBoard:FARDETECT ~= 'NIL
THEN (PRINTOUT T T "These C3 sites lie within 10 degrees" T
 "of the reported heading: " (CONVERT TRENDBoard:FARDETECT)
 T "Threat to these sites is: POSSIBLE." T);
```

```
IF FACTOR = 'DetectBoard
   TRENDBoard:FARDETECT = 'NIL
   TRENDBoard:CLOSEDETECT = 'NIL
   TRENDBoard:WILLDETECT = 'NIL
THEN (PRINTOUT T T "Reported HEADING is further than" T  "1
0 degrees from any C3 site." T "Threat to these sites is: U
NLIKELY." T);

IF FACTOR = 'MATCHBoard
THEN (PRINTOUT T T "Reported LAUNCH SITE is: " MATCHBoard:L
AUNCHSITE T "with " MATCHBoard:LAUNCHERTYPE " missiles." T
T "The CAPABILITY of the Reporting Sensor is: " MATCHBoard:
RPTSENSORCAP T);

IF FACTOR = 'MATCHBoard
   MATCHBoard:OUTAGEREASON ~= 'NIL
THEN (PRINTOUT T "Due to: " MATCHBoard:OUTAGEREASON T);

IF FACTOR = 'TRENDBoard
THEN (PRINTOUT T "Confidence of the ADS-1 Message: " (GETPR
OP EVENTMATCH 'P1CONFIDENCE) T "Confidence of the ADS-2 Mes
sage: " (GETPROP EVENTMATCH 'P2CONFIDENCE) T T);

IF FACTOR = 'TRENDBoard
THEN (SETQ SENSORNUM (GETPROP EVENTMATCH 'P1SENSOR))
      (RunRS 'SensorMap)
      (PRINTOUT T "The Reported Sensor CAPABILITY on the FIR
ST Pass is: " SENSORCAP T "The Sensor CONFIDENCE Level repo
rted on the" T "FIRST pass is: " SENSORCONF T T);

IF FACTOR = 'TRENDBoard
THEN (SETQ SENSORNUM (GETPROP EVENTMATCH 'P2SENSOR))
      (RunRS 'SensorMap)
      (PRINTOUT T "The Reported SECOND PASS Sensor CAPABILIT
Y is: " SENSORCAP T "The SECOND PASS Sensor CONFIDENCE  Lev
el is: " SENSORCONF T T)  OUTVAR←'NIL  MARK←'TYPE2;

(* This rule triggered only by the additional information p
rovided in a BSS message which gives a P3SENSOR.)

IF FACTOR = 'TRENDBoard
   SENSORTYPE = 'BSS
THEN (SETQ SENSORNUM P3SENSOR)
      (RunRS 'SensorMap)    OUTVAR←'T
      (PRINTOUT T "The Reported BSS SITE Sensor CAPABILITY i
s: " SENSORCAP T "The BSS SITE Sensor CONFIDENCE Level is:
" SENSORCONF T T);

IF FACTOR = 'TRENDBoard
```

```
IF FACTOR = 'TRENDBoard
THEN (PRINTOUT T "The Infrared Intensity Reading reported o
n" T "the first pass is: " (GETPROP EVENTMATCH 'P1IRTYPE) T
);

IF FACTOR = 'TRENDBoard
   (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN (PRINTOUT T "Second Pass IR Reading: CONFIRMS" T
"the first pass reading." T);

IF FACTOR= 'TRENDBoard
   (GETPROP EVENTMATCH 'P2TYPE) = 'NOMATCH
THEN (PRINTOUT T "Second Pass IR Reading: IMPROVES" T
"over the first pass reading." T);

IF FACTOR = 'TRENDBoard
   (GETPROP EVENTMATCH 'P2TYPE) = 'MISMATCH
THEN (PRINTOUT T "Second Pass IR Reading: DENIES" T
"the first pass reading." T);

(* Only modification of BSS explanation of TRENDBoard to IR
 TYPE involves further certainty about the LAUNCHER    TYPE
. When all conditions below are met= TYPE1.)

IF SENSORTYPE = 'BSS
   P3STABILITY = 'Y
   P3OBJNUM < 2
   (GETPROP EVENTMATCH 'P1IRTYPE) ~= 'TYPE2
THEN MARK←'TYPE1;

(* The following is the BSS site contribution to identifyin
g the missile type. Note that OUTVAR=T is intended to trigg
er only when explaining BSS events. Only chance for revisio
n is in the third rule below.)

IF OUTVAR = 'T
   MATCHBoard:LAUNCHERTYPE = MARK
THEN (PRINTOUT T T "BSS Site Observations CONFIRM the missi
le type." T);

IF OUTVAR = 'T
   MATCHBoard:LAUNCHERTYPE ~= MARK
THEN (PRINTOUT T "BSS Site Observations DENY the previous m
issile type." T)
MATCHBoard:LAUNCHERTYPE←MARK;

IF FACTOR = 'TRENDBoard
THEN (PRINTOUT T T "SENTINEL concludes the launcher type is
: " MATCHBoard:LAUNCHERTYPE T);
```

```
IF FACTOR = 'TRENDBoard
THEN (PRINTOUT T T "SENTINEL concludes the launcher type is
: " MATCHBoard:LAUNCHERTYPE T);

(* The output of the BSSBoard encompasses the THREAT to
 both CS Node and US targets.)

IF FACTOR = 'BSSBoard
   BSSBoard:WILLDETECT ~= 'NIL
THEN (PRINTOUT T T "The following Targets lie within 1 degr
ee of the reported heading: " (CONVERT BSSBoard:WILLDETECT)
 T "THREAT to these sites is: ALMOST CERTAIN." T);

IF FACTOR = 'BSSBoard
   BSSBoard:CLOSEDETECT ~= 'NIL
THEN (PRINTOUT T T "The following Targets lie within 5 degr
ees of the reported heading: " (CONVERT BSSBoard:CLOSEDETEC
T) T "THREAT to these sites is: LIKELY." T);

IF FACTOR = 'BSSBoard
   BSSBoard:FARDETECT ~= 'NIL
THEN (PRINTOUT T T "The following Targets lie within 10 deg
rees of the reported heading: " (CONVERT BSSBoard:FARDETECT
) T "THREAT to these sites is: POSSIBLE." T);

IF FACTOR = 'BSSBoard
   BSSBoard:FARDETECT = 'NIL
   BSSBoard:WILLDETECT = 'NIL
   BSSBoard:CLOSEDETECT = 'NIL
THEN (PRINTOUT T T "The object tracked has a reported headi
ng of: " P3HEADING T "NO THREAT reported to any US Target."
 T);
#&(RuleSetSource "SPU0.zI[:.;d9.6")
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←(REDO 97)
```

using $EXTHIGHConfidence

RuleSet EXTHIGHConfidence

Last Edited: 3-Oct-85 18:20:54 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   MARK READ WIN1 LOS1 HYPO1 HYPO2;
Control Structure: DOALL;
********



(* EXTHIGHConfidence invoked from ADS2Confidence, takes vot
e to insure confidence is EXTHIGH. Revised to HIGH when nec
essary. Uses Analyze to determine when a special case such
as FIRSTSTRIKE may be ocurring explains output.)
THEN (SETQ TRANSFER 'NIL)
     (RunRS 'BoardVoting);

(* Large LOWVOTE results in belief revision, TRANSFER is  s
et to T to insure belief revision only occurs once.)
IF LOWVOTE >= HIGHVOTE
THEN (SETQ TRANSFER 'T)
   (PRINTOUT T "Confidence estimate downgraded to: HIGH" T)
   (RunRS 'HIGHConfidence) (STOP);

(* HIGHRanking is run to set the FACTORLIST to explain)

THEN (SETQ FACTORLIST (LIST MATCHBoard:EXPNUMBER 'MATCHBoar
d DetectBoard:EXPNUMBER 'DetectBoard TRENDBoard:EXPNUMBER '
TRENDBoard IntelBoard:EXPNUMBER 'IntelBoard SYSBoard:EXPNUM
BER 'SYSBoard));

(* P2CONFIDENCE updated on event list when necessary.)

THEN  (PRINTOUT T "EXTREMELY HIGH Confidence Levels Confirm
ed." T)
 (PUTPROP EVENTMATCH 'P2CONFIDENCE 'EXTHIGH)
 (RunRS 'HIGHRanking)
 (RunRS 'Speculate)
 (RunRS 'HOSTILE)
 (RunRS 'TESTHYPO) MARK←'NIL;

(* Following the running of HOSTILE and TESTHYPO above, the
 leading hypothesis HYPO1 and HYPO2 are determined.)

IF HOSTILENUM >= TESTNUM
THEN HYPO1←'HOSTILE
     HYPO2←'TEST  MARK←'T;

IF MARK = 'NIL THEN HYPO1←'TEST HYPO2←'HOSTILE;
```

```
THEN (RunRS 'UNRESOLVED) MARK←'NIL WIN1←(MKATOM (CONCAT HYP
O1 'NUM))  LOS1←(MKATOM (CONCAT HYPO2 'NUM));

IF UNRESOLVEDNUM > (EVAL WIN1)THEN  HYPO2←HYPO1
      HYPO1←'UNRESOLVED  MARK←'T;

 IF MARK = 'NIL
   UNRESOLVEDNUM > (EVAL LOS1) THEN HYPO2←'UNRESOLVED;

THEN (RunRS 'UNKNOWN) MARK←'NIL
     WIN1←(MKATOM (CONCAT HYPO1 'NUM))
     LOS1←(MKATOM (CONCAT HYPO2 'NUM));

IF UNKNOWNNUM > (EVAL WIN1) THEN HYPO2←HYPO1  HYPO1←'UNKNOW
N MARK←'T;

 IF MARK = 'NIL UNKNOWNNUM > (EVAL LOS1)
THEN HYPO2←'UNKNOWN;

 (* At the end of this section, leading hypo=HYPO1, with
the contender HYPO2 checked for closeness.)THEN MARK←(DIFFE
RENCE (EVAL WIN1) (EVAL LOS1));

IF MARK >= 3 THEN (PUTPROP EVENTMATCH 'P2HYPO HYPO1);

IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL HYPO1 = 'HOSTILE THE
N MARK←'T;

IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL HYPO2 = 'HOSTILE THE
N MARK←'T;

 (* Hypothesis set to HOSTILE in case of doubt when THREAT
is high, other hypos do not discount THREAT.)IF MARK = 'T T
HREATNUM >= 2 NOTHREATNUM < 2 THEN (PUTPROP EVENTMATCH 'P2H
YPO 'HOSTILE);

(* When all else fails select leading hypo.)IF (GETPROP EVE
NTMATCH 'P2HYPO) = 'NIL THEN (PUTPROP EVENTMATCH 'P2HYPO HY
PO1);

 (* OUTPUT SECTION)THEN (PRINTOUT T T " ADS PASS 2 " EVENTM
ATCH T T "1. DETECTED: " PRESENTIME  T) READ←'NIL;

 IF SENSORNUMBER = 'S2 THEN READ←'SOUTH;

 IF READ = 'NIL THEN READ←'NORTH;

 THEN (PRINTOUT T "2. ADS: " READ T "3. EVENT TYPE: " (GETP
ROP EVENTMATCH 'P2HYPO) T "4. BSS DETECT: " DetectBoard:DET
```

```
THEN (PRINTOUT T "2. ADS: " READ T "3. EVENT TYPE: " (GETP
ROP EVENTMATCH 'P2HYPO) T "4. BSS DETECT: " DetectBoard:DET
ECTPT T) READ←'YES;

 IF (GETPROP EVENTMATCH 'P2HYPO) ~= 'HOSTILE THREATNUM < 2
THEN READ←'NO;

(* Special case:when hypothesis revealed is TEST want to co
nvince MWO evidence supports this: Therefore we run    LOWR
anking to put the most conclusive differences.)
 IF (GETPROP EVENTCOUNT 'P2HYPO) = 'TEST
 THEN READ←'NO (SETQ FACTORLIST (LIST MATCHBoard:EXPNUMBER
'MATCHBoard DetectBoard:EXPNUMBER 'DetectBoard TRENDBoard:E
XPNUMBER 'TRENDBoard IntelBoard:EXPNUMBER 'IntelBoard SYSBo
ard:EXPNUMBER 'SYSBoard)) (RunRS 'LOWRanking);

THEN (PRINTOUT T "5. THREAT: " READ T "6. CONFIDENCE:  EXTH
IGH " T  T);

(* Where the running of Speculate above triggers a        s
pecial case the output shows up now.)

IF EXPLAINBoard:COMMENT ~= 'NIL
THEN (EVAL  EXPLAINBoard:COMMENT);

THEN (PRINTOUT T T "If you wish further explanation type T"
T "Otherwise, type NIL to Proceed." T)
    (SETQ READ (RATOM NIL NIL));

 IF READ = 'NIL THEN (STOP);

 THEN (RunRS 'Explain);
#&(RuleSetSource "SIU0.zI[:.Je8.9")
2←
NIL
2←
NIL
2←
NIL
2←
NIL
2←
NIL
2←
NIL
2←
NIL
2←
NIL
2←(REDO 99)
```

using $EXTLOWConfidence

RuleSet EXTLOWConfidence

```
Last Edited: 3-Oct-85 18:25:43 by
WorkSpace Class: Boards;
Compiler Options: ;
Temporary Vars:   MARK READ HYPO1 HYPO2 WIN1 LOS1;
Control Structure: DOALL;
********


(* EXTLOWConfidence invoked from ADS2Confidence, takes vote
 to insure confidence is EXTLOW. Then evaluates the comment
s from the first three factors to explain to the user why t
he report should not be processed. User is asked for his ap
proval to delete or process the report.)
THEN (SETQ TRANSFER 'NIL)
     (RunRS 'BoardVoting);

(* Large HIGHVOTE results in belief revision, TRANSFER is s
et to T to avoid extraneous comments from LOWConfid)
IF HIGHVOTE >= LOWVOTE
THEN (SETQ TRANSFER 'T)
   (PRINTOUT T "Confidence estimate revised to: LOW" T)
   (RunRS 'LOWConfidence) (STOP);

(* LOWRanking is run to set the FACTORLIST to explain)
THEN (SETQ FACTORLIST (LIST DetectBoard:EXPNUMBER 'DetectBo
ard SYSBoard:EXPNUMBER 'SYSBoard IntelBoard:EXPNUMBER 'Inte
lBoard TRENDBoard:EXPNUMBER 'TRENDBoard MATCHBoard:EXPNUMBE
R 'MATCHBoard));

(* Try to convince the USER to delete the report.)
THEN (RunRS 'LOWRanking)
   (PRINTOUT T T "CONFIDENCE LEVELS are extremely LOW" T "
due to the following factors:" T T);

THEN (EVAL (GetValue (EVAL (GetValue EXPLAINBoard 'FACTOR1)
) 'COMMENT))
     (EVAL (GetValue (EVAL (GetValue EXPLAINBoard 'FACTOR2)
) 'COMMENT))
     (EVAL (GetValue (EVAL (GetValue EXPLAINBoard 'FACTOR3)
) 'COMMENT));

THEN (PRINTOUT T T "SENTINEL DECISION AID recommends that t
his report not be processed further due to:" T);

(* Additional factors set according to the 1st FACTOR)
IF EXPLAINBoard:FACTOR1 = 'SYSBoard
THEN (PRINTOUT T "Serious SYSTEM outages." T T);
```

```
IF EXPLAINBoard:FACTOR1 = 'DetectBoard
THEN (PRINTOUT T "No possibilities for further detection."
T T);

IF EXPLAINBoard:FACTOR1 = 'TRENDBoard
THEN (PRINTOUT T "Degrading evidence, spurious sensor repor
t suspected." T T);

IF EXPLAINBoard:FACTOR1 = 'MATCHBoard
THEN (PRINTOUT T "Unreliable sensor report due to damage."
T T);

IF EXPLAINBoard:FACTOR1 = 'IntelBoard
THEN (PRINTOUT T "No indications of any hostile act." T T);

 (* User is asked to signal his approval or disapproval) TH
EN (PRINTOUT T T "If you concur with SENTINEL's reasoning,
" T "Type T to TERMINATE the processing of this event." T "
Otherwise, type NIL to continue processing." T)(SETQ READ (
RATOM NIL NIL));

(* Termination point for further processing.)
IF READ = 'T
THEN (STOP);

(* The following determines the first and second hypothesis
. HYPO1 is the closest with factor number WIN1, while HYPO2
 is second with LOS1. In case of a tie, priority    is HOST
ILE>TEST>UNRESOLVED>UNKNOWN.)

THEN (RunRS 'HOSTILE) (RunRS 'TESTHYPO) MARK←'NIL;

IF HOSTILENUM >= TESTNUM
THEN HYPO1←'HOSTILE
     HYPO2←'TEST   MARK←'T;

IF MARK = 'NIL
THEN HYPO1←'TEST
     HYPO2←'HOSTILE;

THEN (RunRS 'UNRESOLVED) MARK←'NIL
     WIN1←(MKATOM (CONCAT HYPO1 'NUM))
     LOS1←(MKATOM (CONCAT HYPO2 'NUM));

IF UNRESOLVEDNUM > (EVAL WIN1)
THEN HYPO2←HYPO1
     HYPO1←'UNRESOLVED   MARK←'T;

IF MARK = 'NIL
```

```
     HYPO1←'UNRESOLVED   MARK←'T;

IF MARK = 'NIL
   UNRESOLVEDNUM > (EVAL LOS1)
THEN HYPO2←'UNRESOLVED;

THEN (RunRS 'UNKNOWN) MARK←'NIL
     WIN1←(MKATOM (CONCAT HYPO1 'NUM))
     LOS1←(MKATOM (CONCAT HYPO2 'NUM));

IF UNKNOWNNUM > (EVAL WIN1)
THEN HYPO2←HYPO1
     HYPO1←'UNKNOWN   MARK←'T;

IF MARK = 'NIL
   UNKNOWNNUM > (EVAL LOS1)
THEN HYPO2←'UNKNOWN;

(* In this section, heuristics help determine whether      l
eading hypothesis indeed stays as the leading hypo.)

THEN MARK←(DIFFERENCE (EVAL WIN1) (EVAL LOS1))
    (PUTPROP EVENTMATCH 'P2CONFIDENCE 'EXTLOW);

IF MARK >= 3
THEN (PUTPROP EVENTMATCH 'P2HYPO HYPO1);

IF HYPO1 = 'UNKNOWN
THEN (PUTPROP EVENTMATCH 'P2HYPO 'UNKNOWN);

IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
    HYPO1='UNRESOLVED
THEN MARK←'T;

IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
    HYPO2='UNRESOLVED
THEN MARK←'T;

(* Hypothesis set to HOSTILE when THREAT is HIGH and not
 discounted by NOTHREATNUM.)

IF MARK = 'T
   THREATNUM < 2
THEN (PUTPROP EVENTMATCH 'P2HYPO 'UNRESOLVED);

(* When all else fails, select leading hypo as answer.)
IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
THEN (PUTPROP EVENTMATCH 'P2HYPO HYPO1);

(* OUTPUT SECTION)
```

```
(* OUTPUT SECTION)
THEN (PRINTOUT T T "ADS PASS 2 " EVENTMATCH T T "1. DETECTE
D: " PRESENTIME T) READ←'NIL;

IF SENSORNUMBER = 'S2
THEN READ←'SOUTH;

IF READ = 'NIL
THEN READ←'NORTH;

THEN (PRINTOUT T "2. ADS: " READ T "3. EVENT TYPE: " (GETPR
OP EVENTMATCH 'P2HYPO) T "4. BSS DETECT: " DetectBoard:DETE
CTPT T) READ←'NO;

(* Since confidence is EXTLOW, THREAT is assumed to be   NO
 unless proved otherwise.)

IF (GETPROP EVENTMATCH 'P2HYPO) = 'HOSTILE
THEN (SETQ FACTORLIST (LIST SYSBoard:EXPNUMBER 'SYSBoard De
tectBoard:EXPNUMBER 'DetectBoard TRENDBoard:EXPNUMBER 'TREN
DBoard MATCHBoard:EXPNUMBER 'MATCHBoard IntelBoard:EXPNUMBE
R 'IntelBoard))
 (RunRS 'HIGHRanking)
     READ←'YES;

IF (GETPROP EVENTMATCH 'P2HYPO) = 'UNRESOLVED
   THREATNUM >= 2
THEN READ←'YES;

THEN (PRINTOUT T "5. THREAT: " READ T "6. CONFIDENCE: EXTLO
W" T T T "Further explanation can be accessed by typing in
T" T "Otherwise, type NIL to proceed." T)
  (SETQ READ (RATOM NIL NIL));

IF READ = 'T
THEN (RunRS 'Explain);

THEN (PRINTOUT T  "THis is HYPO1: " HYPO1 T "This is HYPO2:
 " HYPO2 T);
#&(RuleSetSource "SIU0.zI[:.Je8.10")
23←
NIL
23←
NIL
23←
NIL
23←
NIL
23←(REDO 99)
```

```
NIL
30←(PP FAN)

(FAN
  [LAMBDA (LAUNCHSITE)                        (* edited: "21-Oct-85 14:40")

              (* FAN; Given a Launchsite, FAN calls on POLAR repeatedly with the
 target and launch coordinates, to put together a FANLIST of the range and headin
g from the launchsite to each member of the TARGETLIST. INPUT: the LAUNCHSI
TE, and its property list. OUTPUT; the FANLIST with (target heading range) eleme
nts)


    (PROG (X Y TGTLIST A)
          (SETQ X (GETPROP LAUNCHSITE (QUOTE LAT)))
          (SETQ Y (GETPROP LAUNCHSITE (QUOTE LONG)))
          (SETQ LNCHCOORD (LIST X Y))
          (SETQ TGTLIST TARGETLIST)
          [SETQ FANLIST (LIST (CONS LAUNCHSITE
                                    (LIST (QUOTE HEADING)
                                          (QUOTE RANGE]
                                           (* Calls on POLAR for each mem
ber of the TARGETLIST)
      LOOP(COND
             ((NULL (CAR TGTLIST))
               (RETURN)))
          (SETQ TARGET (CAR TGTLIST))
          [SETQ TGTCOORD (LIST (GETPROP TARGET
                                        (QUOTE LAT))
                               (GETPROP TARGET
                                        (QUOTE LONG]
          (printout T "THIS IS POLARCALL " TGTCOORD
                    LNCHCOORD T)
          (printout T "THIS IS TARGET NUMBER " TARGET T T)
          (POLAR TGTCOORD LNCHCOORD)
          (SETQ A (LIST TARGET HEADING RANGE))
          (DOCOLLECT A FANLIST)
          (SETQ TGTLIST (CDR TGTLIST))
          (GO LOOP])
(FAN)
31←
NIL
31←
NIL
31←
NIL
31←
NIL
31←
NIL
31←(REDO 99)
```

using $HeadingCheck

RuleSet HeadingCheck

```
Last Edited: 3-Oct-85 18:41:44 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   TLIST TARGET MAXPT MINPT MATCHPT MAXDIFF
MINDIFF BSSNEARDIFF BSSNEARPT BSSWILLDETECT BSSCLOSEDETECT
BSSFARDETECT MARK BSSDETECTPT MIDPOINT MIDIFF OLDMIDIFF;
Control Structure: WHILEALL;
Iteration Condition:   T;
********


(* HeadingCheck goes through BSSTAG TAGLIST and a HEADING t
o compute which BSS site will detect the launch. Those that
 will directly=WILLDETECT, those within 5 degrees =CLOSEDET
ECT, those within 10 FARDETECT. The closest site that is fu
rther than 10 degrees away in coverage is stored in NEARPT
and the difference in NEARDIFF. The BSS site given in the A
DS2 report is DETECTPT. Most of the DetectBoard is filled o
ut here.)

(* One time initialization)

{1!} THEN TLIST←(CDR TAGLIST)
     BSSNEARDIFF←70
     DetectBoard:WILLDETECT←'NIL
     DetectBoard:CLOSEDETECT←'NIL
     DetectBoard:FARDETECT←'NIL
     DetectBoard:NEARPT←'NIL
     DetectBoard:NEARDIFF←'NIL
     DetectBoard:DETECTPT←0
     OLDMIDIFF←70;

IF (CAR TLIST) = 'NIL
THEN DetectBoard:WILLDETECT←BSSWILLDETECT
     DetectBoard:CLOSEDETECT←BSSCLOSEDETECT
     DetectBoard:FARDETECT←BSSFARDETECT
     DetectBoard:NEARPT←BSSNEARPT
     DetectBoard:NEARDIFF←BSSNEARDIFF
     DetectBoard:DETECTPT←(GLC BSSDETECTPT)
     DetectBoard:STATUSDETECTPT←(GETPROP BSSDETECTPT
      'CAPABILITY)
     (STOP);

THEN MATCHPT←(CAR TLIST)
     TARGET←(CAR MATCHPT)
     MAXPT←(CADR MATCHPT)
     MINPT←(CADDR MATCHPT)
```

```
(* Heading between MINPT and MAXPT = WILLDETECT, MARK is se
t to preclude same site joining more than one list.)
IF HEADING <= MAXPT
   HEADING >= MINPT
THEN (SETQ BSSWILLDETECT (APPEND (LIST TARGET) BSSWILLDETEC
T)) MARK←'T;

(* Heading deviation of 5 degrees or less=CLOSEDETECT)
IF MARK = 'NIL
   MAXDIFF <= 5
THEN (SETQ BSSCLOSEDETECT (APPEND (LIST TARGET) BSSCLOSEDET
ECT)) MARK←'T;

IF MARK = 'NIL
   MINDIFF <= 5
THEN (SETQ BSSCLOSEDETECT (APPEND (LIST TARGET) BSSCLOSEDET
ECT)) MARK←'T;

(* Heading deviation of 10 or less degrees=FARDETECT)
IF MARK = 'NIL
   MAXDIFF <= 10
THEN (SETQ BSSFARDETECT (APPEND (LIST TARGET) BSSFARDETECT)
) MARK←'T;

IF MARK = 'NIL
   MINDIFF <= 10
THEN (SETQ BSSFARDETECT (APPEND (LIST TARGET) BSSFARDETECT)
)  MARK←'T;

(* Rule determines which is the best to detect by examining
 deviation from MIDPOINT, Best wins and is DETECTPT)
IF OLDMIDIFF > MIDIFF
THEN BSSDETECTPT←TARGET
     OLDMIDIFF←MIDIFF;

(* For sites with coverage more than 10 degrees away NEARPT
 is set to the site with closest deviation NEARDIFF)
IF   MARK = 'NIL
     BSSNEARDIFF > MINDIFF
THEN BSSNEARPT←TARGET
     BSSNEARDIFF←MINDIFF ;

IF   MARK = 'NIL
     BSSNEARDIFF > MAXDIFF
THEN BSSNEARPT←TARGET
     BSSNEARDIFF←MAXDIFF;

(* To keep the loop going the list is reset to CDR)
THEN (SETQ TLIST (CDR TLIST));
```

```
(HELP
  [LAMBDA NIL                                    (* edited: "10-Oct-85 14:23")
    (PROG (READ)
          (printout T T
"This is the SENTINEL Decision Aid Operator's Manual."
                    T T
          "The basic command for invoking SENTINEL is"
                    T
"(METASCHEDULER MESSAGE) wher MESSAGE is any valid SIMCOPE
message."
                    T
"(MODIFY) can be used after every launch report to modify a
ny of the parameters of the SIMCOPE message under considera
tion."
                    T T
"You cannot use (MODIFY) on SYS or INT reports however."
                    T T
"To initialize SENTINEL and restore actors, events and coun
ters to the proper values type in (SENTINEL). "
                    T T
"To see SENTINEL in operation, there are several example me
ssages which can be used to demonstrate operation:"
                    T
   "For intelligence reports use: INTA, INTB or INTC."
                    T
"For sensor status: SYSMSGA, SYSMSGB, SYSMSGC, or SYSMSGD."
                    T
"For first pass indications: ADS1A, ADS1B, ADS1C, ADS1D, AD
S1U."
                    T
"For further launch reports use: ADS2A, ADS2B, ADS2C and BS
SA, BSSB, and BSSC."
                    T T
"Use them in the following fashion: (METASCHEDULER INTA) an
d so on."
                    T T
"To exit this program type NIL followed by a carriage retur
n <CR> now. Type in T <CR> if you wish to see the troublesh
ooting section."
                    T)
          (SETQ READ (RATOM NIL NIL))
          (COND
            ((NULL READ)
              (RETURN)))
          (printout T
"The following are various troubleshooting aids which have
been helpful in the past."
                    T T
"Have the POLAR, SCENARIO and SENTINEL files been loaded in
```

```
"If not, use the (LOAD 'Filename) command to load in these
files."
                    T T
"The RS232.DCOM, LDI, and FXPRINTER.DCOM files may also be
useful to LOAD in at this time."
                    T
"The rulesets must all be compiled; this is especially true
 when starting up the system from scratch."
                    T T
"Use the (← RuleSetName ER) command first, then use (contro
l) X to fetch the menu. From the menu select OK to compile
the ruleset and return you to LISP."
                    T T
"The rulesets to handle SYS, INT and ADS-1 messages are: MS
GHandler, ADS1Confidence, Regions, ADS1FeatureSet, INTELAna
lyst, INTELComment, Countries, SYSAnalyst, SYSComment, C3An
alyst, SYSCheck, C3Board and Combinatorial."
                    T T
"Insure that the commands featuring rule sets are typed in,
 just as they are above. A change of even one letter or its
 capitalization will result in an UNBOUND statement.")
          (printout T T
"To insure all first pass messages can be handled, type in
(METASCHEDULER INTA) and then use ADS1A and SYSMSGA in plac
e of INTA to verify first pass message processing."
                    T T "Use HELP to see this list again."
                    T T
"If you wish to exit type in NIL now, otherwise type T to p
roceed to second pass message processing."
                    T)
          (SETQ READ (RATOM NIL NIL))
          (COND
            ((NULL READ)
              (RETURN)))
          (printout T T
"To handle ADS-2 messages the following rule sets must be c
ompiled: ADS2Confidence, ADS2FeatureSet, BoardVoting, Expla
in, Explain2, EXTHIGHConfidence, HIGHConfidence, MIXEDConfi
dence, LOWConfidence, EXTLOWConfidence, HIGHRanking, LowRan
king, Speculate, UNKNOWN, HOSTILE, TESTHYPO, UNRESOLVED, Se
nsorCheck, DetectCheck, HeadingCheck, TargetCheck, TrendChe
ck, SensorMap."
                    T T
"To check out ADS2 message processing capabilities, use (ME
TASCHEDULER ADS2A) to verify that all rule sets have been c
ompiled."
                    T T
"When you are ready to exit this portion, type in NIL other
wise type in T to see the rule sets which must be compiled
for BSS message processing."
```

```
just as they are above. A change of even one letter or its
 capitalization will result in an UNBOUND statement.")
          (printout T T
"To insure all first pass messages can be handled, type in
(METASCHEDULER INTA) and then use ADS1A and SYSMSGA in plac
e of INTA to verify first pass message processing."
                    T T "Use HELP to see this list again."
                    T T
"If you wish to exit type in NIL now, otherwise type T to p
roceed to second pass message processing."
                    T)
          (SETQ READ (RATOM NIL NIL))
          (COND
            ((NULL READ)
              (RETURN)))
          (printout T T
"To handle ADS-2 messages the following rule sets must be c
ompiled: ADS2Confidence, ADS2FeatureSet, BoardVoting, Expla
in, Explain2, EXTHIGHConfidence, HIGHConfidence, MIXEDConfi
dence, LOWConfidence, EXTLOWConfidence, HIGHRanking, LowRan
king, Speculate, UNKNOWN, HOSTILE, TESTHYPO, UNRESOLVED, Se
nsorCheck, DetectCheck, HeadingCheck, TargetCheck, TrendChe
ck, SensorMap."
                    T T
"To check out ADS2 message processing capabilities, use (ME
TASCHEDULER ADS2A) to verify that all rule sets have been c
ompiled."
                    T T
"When you are ready to exit this portion, type in NIL other
wise type in T to see the rule sets which must be compiled
for BSS message processing."
                    T T)
          (SETQ READ (RATOM NIL NIL))
          (COND
            ((NULL READ)
              (RETURN)))
          (PRINTOUT T T
"There are only seven rule sets which must be compiled for
BSS message processing: BSSTrendCheck, BSSensorCheck, BSSTG
TCheck, BSSHOSTILE, BSSTEST, BSSUNKNOWN, BSSUNRESOLVED."
                    T T
"Test out the correct operation of SENTINEL for BSS message
s by using the (METASCHEDULER BSSA) command."
                    T T
         "This concludes the troubleshooting section."
                    T])
(HELP)
39←
NIL
39←(REDO 39)
```

using $HIGHConfidence

RuleSet HIGHConfidence

Last Edited: 3-Oct-85 18:22:10 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   READ LOWVOTE HIGHVOTE MARK XVAR YVAR FACT
ORLIST HYPO1 HYPO2 WIN1 LOS1;
Control Structure: DOALL;
********



(* HIGHConfidence invoked from ADS2Confidence, First checks
 to verify CONFIDENCE is HIGH using BoardVoting. Confidence
 will be revised to LOW if LOWVOTE is large enough.Confiden
ce then calls on HIGHRanking to rank order the boards for e
xplanation. HIGHConfidence then runs thorugh the hypothesis
 to decide the ADS-2 Output and explains output to the user
 using two explanation levels with EXPLAIN.)

THEN (RunRS 'BoardVoting)
     XVAR←0 MARK←'NIL;

(* Normally 3 or more votes should be HIGH)

IF HIGHVOTE >= 3
THEN (PRINTOUT T "HIGH Confidence Level Confirmed." T)
     MARK←'T;

(* In cases of ties, blank IntelBoard the HIVALNUMBER in
 TRENDBoard is consulted as tie breaker.)

IF  MARK = 'NIL
    (PLUS HIGHVOTE LOWVOTE) = 4
THEN XVAR←TRENDBoard:HIVALNUMBER;

IF XVAR > 2
THEN HIGHVOTE←(PLUS HIGHVOTE 1);

IF   XVAR > 0
     XVAR <= 2
THEN LOWVOTE←(PLUS LOWVOTE 1);

(* When majority points to LOW belief revision happens.)

IF  LOWVOTE > HIGHVOTE
    TRANSFER = 'NIL
THEN (PRINTOUT T "Confidence Revised to LOW." T)
     (RunRS 'LOWConfidence) (STOP);
```

```
(* For those that didn't make it first time around.)

IF MARK = 'NIL
THEN (PRINTOUT T "HIGH Confidence Level Confirmed." T);

(* The following determines first and second hypothesis  HY
PO1 is the one with the most value, WIN1 and HYPO2 is secon
d leading hypo with LOS1. Priority is             HOSTILE>
TEST>UNRESOLVED>UNKNOWN, ties are updated.)

THEN (RunRS 'HOSTILE) (RunRS 'TESTHYPO) MARK←'NIL;

IF HOSTILENUM >= TESTNUM
THEN HYPO1←'HOSTILE
     HYPO2←'TEST     MARK←'T;

IF MARK = 'NIL
THEN HYPO1←'TEST
     HYPO2←'HOSTILE;

(* Takes advantage that the variable ending to allnumber va
riables which are the number of factors trigerred in  each
hypothesis to place these values in WIN1 and LOS1)
THEN (RunRS 'UNRESOLVED) MARK←'NIL
     WIN1←(MKATOM (CONCAT HYPO1 'NUM))
     LOS1←(MKATOM (CONCAT HYPO2 'NUM));

IF UNRESOLVEDNUM > (EVAL WIN1)
THEN HYPO2←HYPO1
     HYPO1←'UNRESOLVED    MARK←'T;

IF MARK = 'NIL
   UNRESOLVEDNUM > (EVAL LOS1)
THEN HYPO2←'UNRESOLVED;

THEN (RunRS 'UNKNOWN)  MARK←'NIL
     WIN1←(MKATOM (CONCAT HYPO1 'NUM))
     LOS1←(MKATOM (CONCAT HYPO2 'NUM));

IF UNKNOWNNUM > (EVAL WIN1)
THEN HYPO2←HYPO1
     HYPO1←'UNKNOWN  MARK←'T;

IF MARK = 'NIL
   UNKNOWNNUM > (EVAL LOS1)
THEN HYPO2←'UNKNOWN;

(* Now the difference in ranking between HYPO1 and HYPO2  i
s examined to see whether it is significant.)
```

```
THEN MARK←(DIFFERENCE (EVAL WIN1) (EVAL LOS1))
     (PUTPROP EVENTMATCH 'P2CONFIDENCE 'HIGH);

IF MARK >= 3
THEN (PUTPROP EVENTMATCH 'P2HYPO HYPO1);

IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
   HYPO1 = 'HOSTILE
THEN MARK←'T;

IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
   HYPO2 = 'HOSTILE
THEN MARK←'T;

(* When HOSTILE is HYPO1 or HYPO2, difference<3, then    ex
amine THREATNUMBER and NOTHREATNUMBER to decide       EVENT
 is hostile.)
IF MARK = 'T
   THREATNUM >= 2
   NOTHREATNUM < 2
THEN (PUTPROP EVENTMATCH 'P2HYPO 'HOSTILE);

(* When all else fails, select HYPO1 to be it.)
IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
THEN (PUTPROP EVENTMATCH 'P2HYPO HYPO1);

(* OUTPUT SECTION)
THEN (PRINTOUT T T "ADS PASS 2 " EVENTMATCH T T "1. DETECTE
D: " PRESENTIME T)    READ←'NIL;

IF SENSORNUMBER = 'S2
THEN READ←'SOUTH;

IF READ = 'NIL
THEN READ←'NORTH;

THEN (PRINTOUT T "2. ADS: " READ T "3. EVENT TYPE: " (GETPR
OP EVENTMATCH 'P2HYPO) T "4. BSS DETECT: " DetectBoard:DETE
CTPT T)     READ←'YES;

(* Since we are in HIGH confidence range, THREAT will be as
sumed to be YES unless specific rules below trigger.)
IF (GETPROP EVENTMATCH 'P2HYPO) ~= 'HOSTILE
   THREATNUM < 2
THEN READ←'NO;

IF (GETPROP EVENTMATCH 'P2HYPO) = 'TEST
THEN READ←'NO;
```

```
IF (GETPROP EVENTMATCH 'P2HYPO) = 'TEST
THEN READ←'NO;

THEN (PRINTOUT T "5. THREAT: " READ T "6. CONFIDENCE: HIGH"
 T T T " If you wish further explanation type T," T "Otherw
ise, type NIL to Proceed." T)
FACTORLIST←(LIST DetectBoard:EXPNUMBER 'DetectBoard MATCHBo
ard:EXPNUMBER 'MATCHBoard IntelBoard:EXPNUMBER 'IntelBoard
TRENDBoard:EXPNUMBER 'TRENDBoard SYSBoard:EXPNUMBER 'SYSBoa
rd)
   (SETQ READ (RATOM NIL NIL));

IF READ = 'NIL
THEN (STOP);

(* When HYPO is TEST, LOWRank is better way to explain.)
IF (GETPROP EVENTMATCH 'P2HYPO) = 'TEST
THEN (RunRS 'LOWRanking)    MARK←'Y;

IF MARK ~= 'Y
THEN (RunRS 'HIGHRanking);

THEN (PRINTOUT T  "This is FACTOR1: " EXPLAINBoard:FACTOR1
T "This is FACTOR2: " EXPLAINBoard:FACTOR2 T "this is FACTO
R3: " EXPLAINBoard:FACTOR3 T "this is Factor4: " EXPLAINBoa
rd:FACTOR4 T "this is Factor5: " EXPLAINBoard:FACTOR5 T)
  (RunRS 'Explain);
#&(RuleSetSource "SIU0.zI[:.Je8.12")
60←
NIL
60←
NIL
60←
NIL
60←
NIL
60←
NIL
60←
NIL
60←
NIL
60←
NIL
60←
NIL
60←
NIL
60←(REDO 99)
```

Using $HIGHRanking

RuleSet HIGHRanking

Last Edited: 3-Oct-85 18:33:20 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   FOUND FINDSUM FCOUNTER FACTORNUM STOPMARK
;
Control Structure: WHILEALL;
Iteration Condition:   T;
********
```

```
(* HIGHRanking uses a WHILE LOOP to examine each Board's EX
PNUMBER from the given FACTORLIST. Each FACTORLIST should b
e set in the appropriate Confidence Ruleset. HIGHRanking th
en orders them by number. In this case the higher the numbe
r, up to four, the better in case of tie prioritize by plac
e. Results are placed in EXPLAINBoard in the order FACTOR1>
FACTOR2...>FACTOR5 where FACTOR1 would explain first.)

{1!} THEN FCOUNTER←0
          FINDSUM←4
          STOPMARK←6;

(* After initialization, gaps are checked. IntelBoard may b
e blank when no INT reports received. DetectBoard and thus
LSHEADING may be 'XXX when missile heading is not  discerni
ble. Therefore, appropriate spaces on the       EXPLAINBoar
d are set to NIL.)

{1!} IF IntelBoard:EXPNUMBER = 0
   THEN STOPMARK←5
        EXPLAINBoard:FACTOR5←'NIL;

{1!} IF IntelBoard:EXPNUMBER = 0
        MATCHBoard:LSHEADING =  'XXX
   THEN STOPMARK←4
        EXPLAINBoard:FACTOR4←'NIL;

{1!} IF IntelBoard:EXPNUMBER ~= 0
        MATCHBoard:LSHEADING = 'XXX
   THEN STOPMARK←5
        EXPLAINBoard:FACTOR5←'NIL;

(* Iteration starts here, continues with the FCOUNTER until
 it reaches the STOPMARK set above.)
THEN FOUND←(LISTGET1 FACTORLIST FINDSUM);

(* Creates the FACTOR number using a counter.)
```

```
 it reaches the STOPMARK set above.)
THEN FOUND←(LISTGET1 FACTORLIST FINDSUM);

(* Creates the FACTOR number using a counter.)
IF FOUND ~= 'NIL
THEN FCOUNTER←(PLUS FCOUNTER 1)
     FACTORNUM←(CONCAT 'FACTOR FCOUNTER)
     FACTORNUM←(MKATOM FACTORNUM);

(* Once STOPMARK is reached the Rule Set stops.)
IF FCOUNTER >= STOPMARK
THEN (STOP);

(* The appropriate Board is marked into a FACTOR number on
the EXPLAINBoard.)
IF FOUND ~= 'NIL
THEN (PutValue EXPLAINBoard FACTORNUM FOUND)
 (SETQ FACTORLIST (REVERSE FACTORLIST))
 (LISTPUT FACTORLIST FOUND 5)
 (SETQ FACTORLIST (REVERSE FACTORLIST));

(* Having exhausted all the finds for let's say 3, lets loo
k for 2 and then 1.)
IF FOUND = 'NIL
THEN FINDSUM←(SUB1 FINDSUM);

(* Once FINDSUM is less than one the normal range of EXPNUM
BERS has been reached. STOP)
IF FINDSUM < 1
THEN (STOP);
#&(RuleSetSource "SIU0.zI[:.Je8.13")
66←
NIL
66←
NIL
66←
NIL
66←
NIL
66←
NIL
66←
NIL
66←
NIL
66←
NIL
66←
NIL
66←(REDO 99)
```

```
(INTELHANDLER
  [LAMBDA (INTMSG)                          (* edited: "22-Jul-85 14:17")
    (PROG RCXCOUNTRY
      ACTION
      ORGCOUNTRY
      CASUALTYLEVELS
      INTSOURCE
      EQUATION
      INTELMSG)                             (* edited: "18-Jul-85 11:47")
    (SETQ INTELMSG INTMSG)
    (SETQ INTSOURCE (CADR INTELMSG))
    (SETQ CASUALTYLEVELS (CADDAR INTELMSG))
    (SETQ DAYTIME (CADDR (CADAR INTELMSG)))
    (SETQ EQUATION (CADR (CADAR INTELMSG)))
    (SETQ ORGCOUNTRY (CAR EQUATION))
    (SETQ ACTION (CADR EQUATION))
    (SETQ RCXCOUNTRY (CADDR EQUATION))
    (printout T "THIS IS INTSOURCE " INTSOURCE T
              "THIS IS CASUALTYLEVELS "
              CASUALTYLEVELS T "DAYTIME " DAYTIME T
              "ORGCOUNTRY  "
              ORGCOUNTRY T "ACTION " ACTION T
              "RCXCOUNTRY "
              RCXCOUNTRY T)
    (RunRS (QUOTE INTELAnalyst])
(INTELHANDLER)
16←
NIL
16←
NIL
16←(PP HIVALCALL)

(HIVALCALL
  [LAMBDA (LAUNCHSITE)                      (* AMRL " 4-Jun-85 16:50")

          (* Given the HIVALTGTLIST and the LAUNCHSITE, HIVALCALL uses
FAN to produce the high value target list and embeds it in the property list of the l
aunchsite under the name HIVALTAG)


    (PROG NIL
          (SETQ TARGETLIST HIVALTGTLIST)
          (FAN LAUNCHSITE)
          (PUTPROP LAUNCHSITE (QUOTE HIVALTAG)
                   FANLIST)
          (RETURN])
(HIVALCALL)
17←
NIL
17←(REDO 100)
```

using $HOSTILE

RuleSet HOSTILE

Last Edited: 3-Oct-85 18:36:30 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   TNUMBER RNUMBER MARK;
Control Structure: DOALL;
********


(* HOSTILE is a hypothesis rule set that provides a "textbo
ok" example of what a HOSTILE event would result in.  This
example can then be compared to the feature set of the even
t in question and a hypothesis selected.)

THEN TNUMBER←0 RNUMBER←0 MARK←'NIL;

(* TNUMBER records the number of factors that contribute to
 THREAT, RNUMBER number of rules triggered.)

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'TYPE1
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'TYPE2
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GetValue IntelBoard 'INTELCONDITION) = 'RED
THEN RNUMBER←(PLUS RNUMBER 1)
     TNUMBER←(PLUS TNUMBER 1);

IF (GetValue IntelBoard 'INTELCONDITION) = 'YELLOW
THEN RNUMBER←(PLUS RNUMBER 1)
     TNUMBER←(PLUS TNUMBER 1);

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'HIGH
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'EXTHIGH
THEN RNUMBER←(PLUS RNUMBER 1);

IF MATCHBoard:LAUNCHSITE = 'KNOWN
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1SENSOR) >= 7
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN RNUMBER←(PLUS RNUMBER 1);
```

```
IF MATCHBoard:LAUNCHSITE = 'KNOWN
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1SENSOR) >= 7
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2SENSOR) >= 7
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN RNUMBER←(PLUS RNUMBER 1);

IF MATCHBoard:EXPNUMBER >= 3
THEN RNUMBER←(PLUS RNUMBER 1);

IF DetectBoard:EXPNUMBER = 4
THEN RNUMBER←(PLUS RNUMBER 2)
     TNUMBER←(PLUS TNUMBER 1);

IF TRENDBoard:EXPNUMBER >= 3
THEN RNUMBER←(PLUS RNUMBER 1);

IF TRENDBoard:HIVALNUMBER >= 3
THEN RNUMBER←(PLUS RNUMBER 2)
     TNUMBER←(PLUS TNUMBER 1);

THEN (SETQ HOSTILENUM RNUMBER)
     (SETQ THREATNUM  TNUMBER);
#&(RuleSetSource "SWU0.zI[:.<;:.8")
75←
NIL
75←
NIL
75←
NIL
75←
NIL
75←
NIL
75←
NIL
75←
NIL
75←(REDO 99)
```

using $INTELAnalyst

RuleSet INTELAnalyst

```
Last Edited: 3-Oct-85 18:01:57 by
WorkSpace Class: Reports;
Compiler Options:  ;
Temporary Vars:   OLDCOND COUNTRYSOURCE TYPESOURCE;
Control Structure: DOALL;
********




(* This ruleset analyzes incoming INT reports and
   updates the IntelBoard.)

IF ACTION > 0
THEN (SETQ INTTYPE 'POSITIVE);

IF ACTION < 0
THEN (SETQ INTTYPE 'NEGATIVE)
     (SETQ ACTION (ABS ACTION));

IF ACTION = 8
THEN (SETQ CONDITION 'TEST)
     (SETQ ACTION 0);

IF ACTION = 3
THEN (SETQ CONDITION 'RED);

IF ACTION = 4
THEN (SETQ CONDITION 'RED);

IF ACTION = 1
THEN (SETQ CONDITION 'YELLOW);

IF ACTION = 2
THEN (SETQ CONDITION 'YELLOW);

IF ACTION >= 5
THEN (SETQ CONDITION 'GREEN);

(* Negative conditions are processed: All negative
 conditions except -YELLOW result in YELLOW.
 -YELLOW results in GREEN.)

IF INTTYPE = 'NEGATIVE
   CONDITION ~= 'YELLOW
THEN (SETQ INTTYPE 'YELLOW);
```

```
IF INTTYPE = 'YELLOW
THEN (SETQ CONDITION 'YELLOW)
     (SETQ INTTYPE 'NEGATIVE);

IF INTSOURCE >= 0
THEN (SETQ TYPESOURCE 'OPEN);

IF INTSOURCE < 0
THEN (SETQ TYPESOURCE 'INTEL);

(* This section provides the output to the user.)

THEN (SETQ COUNTRYNUM (ABS INTSOURCE))
     (RunRS 'Countries)
     (SETQ COUNTRYSOURCE COUNTRY)
     (SETQ COUNTRYNUM ORGCOUNTRY)
     (RunRS 'Countries) (CF)
(PRINTOUT T "INT message acknowledged:" T TYPESOURCE " sour
ces from the country of: " COUNTRYSOURCE T "reveal existenc
e of CONDITION: " CONDITION  " for the country of: " COUNTR
Y T)
    (PutValue IntelBoard 'INTELDAYTIME DAYTIME);

(* The IntelBoard is updated based on the priority RED>TEST
>YELLOW>GREEN>empty (NIL) in case of a tie, the IntelBoard
is changed for recency.)

IF IntelBoard:INTELCONDITION = 'YELLOW
   CONDITION = 'GREEN
THEN (STOP);

IF IntelBoard:INTELCONDITION = 'TEST
   CONDITION = 'YELLOW
THEN (STOP);

IF IntelBoard:INTELCONDITION = 'TEST
   CONDITION = 'GREEN
THEN (STOP);

IF IntelBoard:INTELCONDITION = 'RED
   CONDITION ~= 'RED
THEN (STOP);

(* This code fills out the IntelBoard)

THEN (PutValue IntelBoard 'SOURCE TYPESOURCE)
     (PutValue IntelBoard 'SOURCECOUNTRY COUNTRYSOURCE)
     (PutValue IntelBoard 'REPORTEDCOUNTRY COUNTRY)
     (PutValue IntelBoard 'INTELTYPE INTTYPE)
(PRINTOUT T "IntelBoard updated. " T);

(* Finally, the COMMENT on the IntelBoard is filled.)
THEN (RunRS 'INTELComment);
#&(RuleSetSource "LVU0.zI[:.P9B.29")
```

Using $INTELComment
RuleSet INTELComment

```
Last Edited: 3-Oct-85 16:02:59 by
WorkSpace Class: Reports;
Compiler Options:  ;
Temporary Vars:   LSVAR LVAR;
Control Structure: DOALL;
********


(* This rule set is invoked from upon receipt of an INT
   message and places an EXPNUMBER and a COMMENT on Intel
   Board, based on INTELCONDITION and SOURCE.)


THEN IntelBoard:EXPNUMBER  ← 0;

IF IntelBoard:INTELCONDITION = 'RED
   IntelBoard:SOURCE = 'INTEL
THEN IntelBoard:EXPNUMBER ← 4;

IF IntelBoard:INTELCONDITION = 'YELLOW
   IntelBoard:SOURCE = 'INTEL
THEN IntelBoard:EXPNUMBER ← 3;

IF IntelBoard:INTELCONDITION = 'RED
   IntelBoard:SOURCE = 'OPEN
 THEN IntelBoard:EXPNUMBER ← 3;

IF IntelBoard:INTELCONDITION = 'YELLOW
   IntelBoard:SOURCE = 'OPEN
THEN IntelBoard:EXPNUMBER ← 2;

IF IntelBoard:INTELCONDITION = 'GREEN
   IntelBoard:SOURCE = 'OPEN
  THEN IntelBoard:EXPNUMBER ← 2;

IF IntelBoard:INTELCONDITION = 'GREEN
   IntelBoard:SOURCE = 'INTEL
THEN IntelBoard:EXPNUMBER ← 1;

(* The Two special conditions cause by TEST with OPEN
   or INTEL source are handled separately.)

IF IntelBoard:INTELCONDITION = 'TEST
   IntelBoard:SOURCE = 'OPEN
THEN IntelBoard:EXPNUMBER ← 2
     IntelBoard:COMMENT ← '(PRINTOUT T T "INTEL reports ind
icate" T "TEST LAUNCH is possible." T )
```

```
IF IntelBoard:INTELCONDITION = 'TEST
   IntelBoard:SOURCE = 'INTEL
THEN IntelBoard:EXPNUMBER ← 3
     IntelBoard:COMMENT ← '(PRINTOUT T T "INTEL conditions
indicate" T "suspected enemy TEST is likely." T) (STOP);

IF IntelBoard:EXPNUMBER = 4
THEN IntelBoard:COMMENT ← '(PRINTOUT T T "INTEL CONDITIONS
STRONGLY SUGGEST" T "IMMINENT ATTACK " T);

IF IntelBoard:EXPNUMBER = 3
THEN IntelBoard:COMMENT ← '(PRINTOUT T T "INTEL reports ind
icate" T "POSSIBLE ATTACK" T);

IF IntelBoard:EXPNUMBER = 2
THEN IntelBoard:COMMENT ← '(PRINTOUT T T "INTEL conditions
suggest" T "ATTACK is NOT likely." T);

IF IntelBoard:EXPNUMBER = 1
THEN IntelBoard:COMMENT ← '(PRINTOUT T T "INTEL reports " T
 "NO indications of ATTACK." T);

THEN (EVAL (GetValue IntelBoard 'COMMENT));
#&(RuleSetSource "GWU0.zI[:.PG8.32")
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←
NIL
10←(REDO 100)
```

using $LOWConfidence

RuleSet LOWConfidence

Last Edited: 3-Oct-85 18:24:12 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   READ LOWVOTE HIGHVOTE MARK XVAR YVAR FACT
ORLIST HYPO1 HYPO2 WIN1 LOS1;
Control Structure: DOALL;
********



(* LOWConfidence invoked from ADS2Confidence, First checks
to verify CONFIDENCE is LOW using BoardVoting. Confidence w
ill be revised to HIGH if HIGHVOTE is large enoughConfidenc
e then calls on LOWRanking to rank order the boards for exp
lanation. LOWConfidence then explains output to the user us
ing two explanation levels with EXPLAIN.)

THEN (RunRS 'BoardVoting)
     XVAR←0 MARK←'NIL;

(* Normally 3 or more votes should be low)

IF LOWVOTE >= 3
THEN (PRINTOUT T "LOW Confidence Level Confirmed." T)
     MARK←'T;

(* When IntelBoard is empty, the number of votes is 4,    c
heck the HIVALNUMBER to resolve the tie.)

IF (PLUS HIGHVOTE LOWVOTE) = 4
THEN XVAR←TRENDBoard:HIVALNUMBER;

IF XVAR >= 3
THEN HIGHVOTE←(PLUS HIGHVOTE 1);

IF XVAR > 0
   XVAR <= 2
THEN LOWVOTE←(PLUS LOWVOTE 1);

(* When the number of HIGHVOTES is higher than for LOW    V
OTES, Belief Revision takes place.)

IF HIGHVOTE > LOWVOTE
   TRANSFER = 'NIL
THEN (PRINTOUT T "Confidence Revised to HIGH." T)
     (RunRS 'HIGHConfidence) (STOP);

(* For those that did not get confirmed until after
```

```
TRENDboard was consulted.)

IF MARK = 'NIL
THEN (PRINTOUT T "LOW Confidence Level Confirmed." T);

(* The following determines what the first and second    le
ading hypothesis are. HYPO1 is the closest with factor numb
er WIN1. HYPO2 in second with LOS1. Prioity  is HOSTILE>TES
T>UNRESOLVED>UNKNOWN.)

THEN (RunRS 'HOSTILE) (RunRS 'TESTHYPO) MARK←'NIL;

IF HOSTILENUM >= TESTNUM
THEN HYPO1←'HOSTILE
     HYPO2←'TEST       MARK←'T;

IF MARK = 'NIL
THEN HYPO1←'TEST
     HYPO2←'HOSTILE;

THEN (RunRS 'UNRESOLVED) MARK←'NIL
     WIN1←(MKATOM (CONCAT HYPO1 'NUM))
     LOS1←(MKATOM (CONCAT HYPO2 'NUM));

IF UNRESOLVEDNUM > (EVAL WIN1)
THEN HYPO2←HYPO1
     HYPO1←'UNRESOLVED    MARK←'T;

IF MARK= 'NIL
   UNRESOLVEDNUM > (EVAL LOS1)
THEN HYPO2←'UNRESOLVED;

THEN (RunRS 'UNKNOWN) MARK←'NIL
     WIN1←(MKATOM (CONCAT HYPO1 'NUM))
     LOS1←(MKATOM (CONCAT HYPO2 'NUM));

IF UNKNOWNNUM > (EVAL WIN1)
THEN HYPO2←HYPO1
     HYPO1←'UNKNOWN  MARK←'T;

IF MARK = 'NIL
   UNKNOWNNUM > (EVAL LOS1)
THEN HYPO2←'UNKNOWN;

(* At the end of this section, leading hypo is embedded  in
 the EVENTLIST after heuristics ascertain difference)

THEN MARK←(DIFFERENCE (EVAL WIN1) (EVAL LOS1))
     (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

     •
```

```
    (PUTPROP EVENTMATCH 'P2CONFIDENCE 'LOW);

IF MARK >= 3
THEN (PUTPROP EVENTMATCH 'P2HYPO HYPO1);

IF HYPO1 = 'UNKNOWN
THEN (PUTPROP EVENTMATCH 'P2HYPO 'UNKNOWN);

IF HYPO1 = 'UNRESOLVED
THEN (PUTPROP EVENTMATCH 'P2HYPO 'UNRESOLVED);

IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
   HYPO2 = 'UNRESOLVED
   THREATNUM < 2
THEN (PUTPROP EVENTMATCH 'P2HYPO 'UNRESOLVED);

IF  (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
    HYPO2 = 'UNKNOWN
    THREATNUM < 2
THEN (PUTPROP EVENTMATCH 'P2HYPO 'UNKNOWN);

(* When all else fails, select leading hypothesis)
IF (GETPROP EVENTMATCH 'P2HYPO) = 'NIL
THEN (PUTPROP EVENTMATCH 'P2HYPO HYPO1);

THEN (PRINTOUT T "this is HYPO1: " HYPO1  T "This is HYPO2
" HYPO2 T);

(* FACTORLIST is now established and permits LOW or HIGHRan
king rule sets to order the boards into an            expla
nation order.)
THEN FACTORLIST←(LIST SYSBoard:EXPNUMBER 'SYSBoard DetectBo
ard:EXPNUMBER 'DetectBoard TRENDBoard:EXPNUMBER 'TRENDBoard
 MATCHBoard:EXPNUMBER 'MATCHBoard IntelBoard:EXPNUMBER 'Int
elBoard);

(* Context dependent reasoning: when HYPO is HOSTILE,    ra
nking is by High numbers, otherwise LOW Ranking.)

IF (GETPROP EVENTMATCH 'P2HYPO) = 'HOSTILE
THEN (RunRS 'HIGHRanking);

IF (GETPROP EVENTMATCH 'P2HYPO) ~= 'HOSTILE
THEN (RunRS 'LOWRanking);

(* OUTPUT SECTION)
THEN (PRINTOUT T T "ADS PASS 2 " EVENTMATCH T T "1. DETECTE
D: " PRESENTIME T) READ←'NIL;

 IF SENSORNUMBER = 'S2 THEN READ←'SOUTH;
```

```
(* OUTPUT SECTION)
THEN (PRINTOUT T T "ADS PASS 2 " EVENTMATCH T T "1. DETECTE
D: " PRESENTIME T) READ←'NIL;

 IF SENSORNUMBER = 'S2 THEN READ←'SOUTH;

 IF READ = 'NIL THEN READ←'NORTH;

 THEN (PRINTOUT T  "2. ADS: " READ T "3. EVENT TYPE: " (GET
PROP EVENTMATCH 'P2HYPO) T "4. BSS DETECT: " DetectBoard:DE
TECTPT T)   READ←'NO;

 (* Since confidence level is LOW NO THREAT assumed.)
 IF (GETPROP EVENTMATCH 'P2HYPO) = 'HOSTILE
 THEN READ←'YES;

 IF (GETPROP EVENTMATCH 'P2HYPO) = 'UNRESOLVABLE
     THREATNUM >= 2
THEN READ←'YES;

 THEN (PRINTOUT T "5. THREAT: " READ T "6. CONFIDENCE: LOW"
 T T T  "Type T and a carriage return for further explanati
on," T "Type NIL to proceed." T) (RunRS 'Speculate) (SETQ R
EAD (RATOM NIL NIL));

 IF EXPLAINBoard:COMMENT ~= 'NIL THEN (EVAL EXPLAINBoard:CO
MMENT);

 IF READ = 'NIL THEN (STOP);

 THEN (PRINTOUT T "This is FACTOR1: " EXPLAINBoard:FACTOR1
T "This is FACTOR2: " EXPLAINBoard:FACTOR2 T "this is FACTO
R3: " EXPLAINBoard:FACTOR3 T "this is Factor4: " EXPLAINBoa
rd:FACTOR4 T "this is Factor5: " EXPLAINBoard:FACTOR5 T)  (
RunRS 'Explain);
#&(RuleSetSource "G↑U0.zI[:.n59.73")
100←
NIL
100←
NIL
100←
NIL
100←
NIL
100←
NIL
100←
NIL
100←(RS232.PRINT (WHICHW) T)
```

RuleSet LOWRanking

Last Edited: 3-Oct-85 18:34:08 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   FOUND FINDSUM FCOUNTER FACTORNUM STOPMARK
;
Control Structure: WHILEALL;
Iteration Condition:   T;
********



(* LOWRanking uses a WHILE LOOP to examine each Board's EXP
NUMBER from the given FACTORLIST. Each FACTORLIST should be
 set in the appropriate Confidence Ruleset. LOWRanking then
 order them by number. In this case the lower the number th
e better in case of tie prioritize by place. Results are pl
aced in EXPLAINBoard in the order FACTOR1>FACTOR2...>FACTOR
5 where FACTOR1 would explain first.)

{1!} THEN FCOUNTER←0
          FINDSUM←1
          STOPMARK←6;

(* After initialization, gaps are checked. IntelBoard   may
 be empty when no INT reports received. LSHEADING may be 'X
XX and DetectBoard will contain no info when no    discerni
ble heading detected for missile.)

{1!} IF IntelBoard:EXPNUMBER = 0
   THEN STOPMARK←5
        EXPLAINBoard:FACTOR5←'NIL;

{1!} IF IntelBoard:EXPNUMBER = 0
        MATCHBoard:LSHEADING = 'XXX
   THEN STOPMARK←4
        EXPLAINBoard:FACTOR4←'NIL;

{1!} IF IntelBoard:EXPNUMBER ~= 0
        MATCHBoard:LSHEADING = 'XXX
   THEN STOPMARK←5
        EXPLAINBoard:FACTOR5←'NIL;

(* Iteration starts here, continues with the FCOUNTER until
 it reaches the STOPMARK set above.)

THEN FOUND←(LISTGET1 FACTORLIST FINDSUM);

(* Creates the FACTOR number using a counter.)
```

```
it reaches the STOPMARK set above.)

THEN FOUND←(LISTGET1 FACTORLIST FINDSUM);

(* Creates the FACTOR number using a counter.)

IF FOUND ~= 'NIL
THEN FCOUNTER←(PLUS FCOUNTER 1)
     FACTORNUM←(CONCAT 'FACTOR FCOUNTER)
     FACTORNUM←(MKATOM FACTORNUM);

(* Once STOPMARK is reached the Rule Set stops.)

IF FCOUNTER >= STOPMARK
THEN (STOP);

(* The appropriate Board is marked into a FACTOR number on
the EXPLAINBoard.)

IF FOUND ~= 'NIL
THEN (PutValue EXPLAINBoard FACTORNUM FOUND)
 (SETQ FACTORLIST (REVERSE FACTORLIST))
 (LISTPUT FACTORLIST FOUND 5)
 (SETQ FACTORLIST (REVERSE FACTORLIST));

(* Having exhausted all the finds for lets say 2, lets look
 for 3 and then 4.)

IF FOUND = 'NIL
THEN FINDSUM←(PLUS FINDSUM 1);

(* Once FINDSUM has exceeded four the normal range of EXPNU
MBERS has been reached. STOP)

IF FINDSUM > 4
THEN (STOP);
#&(RuleSetSource "G↑U0.zI[:.n59.74")
6←
NIL
6←
NIL
6←
NIL
6←
NIL
6←
NIL
6←
NIL
6←(REDO 100)
```

```
(METASCHEDULER
  [LAMBDA (SIMCOPEMSG)                          (* edited: "23-Jul-85 13:08")
    (PROG NIL
          (SETQ MESSAGE SIMCOPEMSG)
          (SETQ DESTINATION (CAAR MESSAGE))
                                                (* SENSORTYPE is equivalent to
ADS, SYS or INT)
          (SETQ SENSORTYPE (CAR (CADAR MESSAGE)))
          (SETQ SENSORNUMBER (CAAADR (CADAR MESSAGE)))
          (RunRS (QUOTE MSGHandler)
                 MESSAGE])
(METASCHEDULER)
13←
NIL
13←
NIL
13←
NIL
13←(PP SYSHANDLER)

(SYSHANDLER
  [LAMBDA (SYSMSG)                              (* edited: "22-Jul-85 16:42")
    (PROG (OUTAGETYPE SENSORCAPABILITY TIMETORESTORE
                      OUTAGETIME SENSORTYPE SUBSYSTEM
                      FRAGMENT REASON SYSTEMSG)
          (SETQ SYSTEMSG SYSMSG)
          (SETQ SENSORCAPABILITY (CADR SYSTEMSG))
          (SETQ TIMETORESTORE (CADDAR SYSTEMSG))
          (SETQ FRAGMENT (CDADAR SYSTEMSG))
          (SETQ OUTAGETYPE (CAAR FRAGMENT))
          (SETQ OUTAGETIME (CADR FRAGMENT))
          (SETQ SENSORTYPE (CADAR FRAGMENT))
          (SETQ SUBSYSTEM (CADDAR FRAGMENT))
          (SETQ REASON (CADR (CDDAR FRAGMENT)))
          (printout T "THIS IS OUTAGE TYPE " OUTAGETYPE T
                    "THIS IS SENSORCAPABILITY "
                    SENSORCAPABILITY T
                    "THIS IS TIMETORESTORE "
                    TIMETORESTORE T "THIS IS OUTAGETIME "
                    OUTAGETIME T "THIS IS SENSORTYPE "
                    SENSORTYPE T "THIS IS SUBSYSTEM  "
                    SUBSYSTEM T "THIS IS REASON " REASON T)
          (RunRS (QUOTE SYSAnalyst])
(SYSHANDLER)
14←
NIL
14←
NIL
14←(REDO 100)
```

```
using $MIXEDConfidence
```

RuleSet MIXEDConfidence

```
Last Edited: 3-Oct-85 18:22:54 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   MARK;
Control Structure: DOALL;
********


(* MIXEDConfidence resolves the ambiguity involved in an AD
S-2 State that results in a CONFIDENCE level of MIXED  by e
xamining the outcome of Board Voting. When neither HIGH nor
 LOW have a majority vote, then TRENDBoard is   consulted:
HIVALNUMBER of 3 or 4 results in HIGH while    a 1 0r 2 res
ults in LOW.)

THEN (RunRS 'BoardVoting)
     MARK←0;

IF HIGHVOTE >= 3
THEN (PRINTOUT T "CONFIDENCE resolved to HIGH." T)
     (RunRS 'HIGHConfidence)
     (STOP);

IF LOWVOTE >= 3
THEN (PRINTOUT T "CONFIDENCE resolved to LOW." T)
     (RunRS 'LOWConfidence)
     (STOP);

(* When there is no simple majority vote, consult the
HIVALNUMBER on the TRENDBoard to decide, otherwise it may g
et routed from MIXED to HIGH and back to LOW.)

IF TRENDBoard:HIVALNUMBER >= 3
THEN (PRINTOUT T "CONFIDENCE resolved to HIGH." T)
     (RunRS 'HIGHConfidence)
     (STOP);

IF TRENDBoard:HIVALNUMBER <= 2
THEN (PRINTOUT T "CONFIDENCE resolved to LOW." T)
     (RunRS 'LOWConfidence)
     (STOP);
#&(RuleSetSource "SIU0.zI[:.Je8.14")
3←
NIL
3←
NIL
3←(REDO 1)
```

```
(MODIFY
  [LAMBDA NIL                                        (* edited: "18-Sep-85 15:14")
    (PROG (READ SENSOR LTRSENSOR)

             (* If SENTINEL was just initialized, cannot use MODIFY since no ME
SSAGE is in the system; otherwise EVENT gets set to EVENTO and NIL are used a
s tags for property lists)

          [COND
            ((NULL MESSAGE)
              (RETURN (printout T T
"Since SENTINEL was just initialized, No MESSAGE is in the
system and no modifications can be done"
                                T]
                                                     (* EVCOUNTER set back by one
so that MODIFICATION will write over EVENT property list rather than creating a n
ew event.)
          (SETQ LTRSENSOR (CAAAR (CDADAR MESSAGE)))
          [COND
            ((EQP (GLC LTRSENSOR)
                  2)
              (RETURN (MODIFY2 MESSAGE]
                                                     (* Pulls off the type of sensor, S1
, N1 or N2, S2, Second pass messages diverted to MODIFY2)
          [COND
            ((EQP (CAR (CADAR MESSAGE))
                  (QUOTE BSS))
              (RETURN (MODIFY3 MESSAGE]
                                                     (* BSS message to be modified i
s sent to MODIFY3)
          [COND
            ((EQP (CAR (CADAR MESSAGE))
                  (QUOTE INT))
              (RETURN (printout T
"Last message received was an INT message which cannot be m
odified."
                                T]
          [COND
            ((EQP (CAR (CADAR MESSAGE))
                  (QUOTE SYS))
              (RETURN (printout T
"Last message was an SYS report which cannot be modified."
                                T]
          (SETQ EVCOUNTER (SUB1 EVCOUNTER))
          (printout T "MESSAGE labeled as " EVENTCOUNT
                    " can now be modified."
                    T T
"Type NIL in response to a question when you wish to exit t
his program"
```

```
 type NIL in response to a question when you wish to exit t
his program"
                      T T
       "Current value for SENSOR SITE CONFIDENCE is: "
                      (CADDR MESSAGE)
                      T T
"The SITE CONFIDENCE can vary from 0 (LOW) to 9 (HIGH)"
                      T
   "type in the new value and hit a carriage return. "
                      T)
           (SETQ READ (RATOM NIL NIL))
           (COND
             ((NULL READ)
               (RETURN)))
           (RPLACA (CDDR MESSAGE)
                   READ)
```

(* Having permitted the modification of the SITE CONFIDENCE, now the IR reading can be reset, Warning: Use of RPLACA as follows modifies the original MESSAGE as well as the variable.)

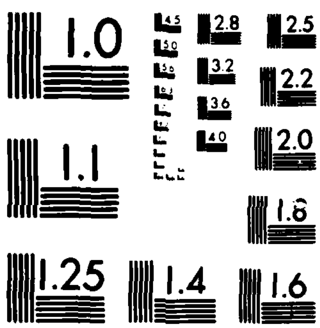
```
           (printout T T
     "Current Value of InfraRed Reading of Sensor is: "
                      (CADAR (CDADAR MESSAGE))
                      T
"Value of this parameter can vary from 0 to 9. Type in the
new value."
                      T)
           (SETQ READ (RATOM NIL NIL))
           [COND
             ((NULL READ)
               (RETURN (METASCHEDULER MESSAGE]
           (RPLACA (CDAR (CDADAR MESSAGE))
                   READ)             (* The following permits the modi
ficaton of the sensor capability)
           [COND
             ((EQ LTRSENSOR (QUOTE S1))
               (SETQ SENSOR (QUOTE ADSSOUTH]
           [COND
             ((EQ LTRSENSOR (QUOTE N1))
               (SETQ SENSOR (QUOTE ADSNORTH]
           (printout T T "Current Capability for sensor: "
                      SENSOR T "is reported as: "
                      (GETPROP SENSOR (QUOTE CAPABILITY))
                      T T
"SENSOR CAPABILITY can be FULL, PARTIAL or ZERO. Type in on
e of these three values followed by a carriage return."
                      T)
           (SETQ READ (RATOM NIL NIL))
           [COND
```

```
          [COND
            ((NULL READ)
              (RETURN (METASCHEDULER MESSAGE]
          (PUTPROP SENSOR (QUOTE CAPABILITY)
                   READ)              (* INTEL condition from the Intel
Board can now be changed)
          (printout T T "Current INTEL Condition is: "
                    (GetValue IntelBoard (QUOTE
                                          INTELCONDITION))
                    T T
        "INTEL Conditions fall into these categories:"
                    T "RED: indicate imminent attack" T
                    "YELLOW: indicate possible attack"
                    T
            "GREEN: no attack indications are present"
                    T
        "TEST: indications of a test launch received."
                    T T
"Type in the new value followed by a carriage return."
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (METASCHEDULER MESSAGE]
          (PutValue IntelBoard (QUOTE INTELCONDITION)
                    READ)             (* INTEL source can be changed
 as well; only possibilities OPEN and INTEL)
          (printout T T "INTEL source is reported to be: "
                    (GetValue IntelBoard (QUOTE SOURCE))
                    T T
"INTEL sources can be OPEN (meaning publicly known or annou
nced) or INTEL (obtained from a classified source). "
                    T
 "Type in the new value followed by a carriage return"
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (METASCHEDULER MESSAGE]
          (PutValue IntelBoard (QUOTE SOURCE)
                    READ)             (* Latitude of the launchsite in th
e MESSAGE modified; BUT NOT the LAT tag on the property list of the identified l
aunchsite.)
          (printout T T
            "Reported Latitude of the Launchsite is: "
                    LNCHLAT " degrees W" T T
"The SIMCOPE scenario deals with latitudes ranging from 10
to 40 degrees. Type in the new value. "
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
```

```
            [COND
              ((NULL READ)
                (RETURN (METASCHEDULER MESSAGE]
            (RPLACA (CDDR (CADAR MESSAGE))
                    READ)                (* Longitude of launchsite descri
bed in the message can be changed.)
            (printout T T
             "Reported longitude of the Launchsite is: "
                      LNCHLONG "degrees S" T T
"The SIMCOPE scenario deals with longitudes ranging from 15
 to 45 degrees."
                      T
 "Type in the new value along with a carriage return."
                      T)
            (SETQ READ (RATOM NIL NIL))
            [COND
              ((NULL READ)
                (RETURN (METASCHEDULER MESSAGE]
            (RPLACA (CDDAR MESSAGE)
                    READ)                (* P1TIME; that is the time when
the ADS-2 reports is expected to arrive can be modified as well.)
            (printout T T
"The expected TIME of arrival for the ADS2 Report of this l
aunch event is: "
                      (CAADR MESSAGE)
                      T T
  "The ADS2 TIME is a six digit number such as 123456"
                      T
       "12 is the number of hours on a 24 hour clock "
                      T "34 is the number of minutes " T
                      "and 56 is the number of seconds."
                      T T
 "Type in the new value along with a carriage return."
                      T)
            (SETQ READ (RATOM NIL NIL))
            [COND
              ((NULL READ)
                (RETURN (METASCHEDULER MESSAGE]
            (RPLACA (CADR MESSAGE)
                    READ)
            (printout T T
       "SENTINEL DECISION AID invoked with new value."
                      T)
            (METASCHEDULER MESSAGE])
(MODIFY)
29←
NIL
29←
NIL
29←(REDO 100)
```

AD-A163 827 AN EXPERT SYSTEM DECISION AID FOR A COMMAND CONTROL AND 4/4
COMMUNICATIONS OPERATOR(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D L TOBAT
UNCLASSIFIED DEC 85 AFIT/GE/ENG/85D-50 F/G 17/2 NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

```
(MODIFY2
  [LAMBDA (MESSAGE)                              (* edited: "19-Sep-85 17:40")
    (PROG (SENSOR READ)

          (* This function does for ADS2 Reports what MODIFY does for ADS1
 Reports; mainly to permit the user to modify various parameters of the MESSAGE
which is then sent back to METASCHEDULER. MODIFY2 is quite similar to MODIF
Y and its code due to the large number of PRINTOUT statments is fairly self docu
menting)

          [COND
            ((NULL EVENTMATCH)
              (RETURN (printout T
                          "There is NO EVENTMATCH: "
                                T
           "SEQUENCE of messages is out of order OR, "
                                T
                   "IF you still wish to use MODIFY,"
                                T
               "SETQ EVENTMATCH to an existing EVENT"
                                T]
          (PUTPROP EVENTMATCH (QUOTE P1TIME)
                   PRESENTIME)
          (printout T T "ADS2 REPORT labeled as "
                    EVENTMATCH " can now be modified." T T
                   "The IR reading for this pass is: "
                    (CADAR (CDADAR MESSAGE))
                    T
            "IR Readings fall into these categories: "
                    T T "1 is Type I" T "2 is Type 2" T
                    "U is Unknown"
                    T "I is Indistinct" T T
"Type in the new value for the IR reading or type NIL to ex
it this program followed by a carriage return."
                    T)
          (SETQ READ (RATOM NIL NIL))
          (COND
            ((NULL READ)
              (RETURN)))
          (RPLACA (CDAR (CDADAR MESSAGE))
                  READ)
          (REMPROP EVENTMATCH (QUOTE P2TYPE))
          (printout T T
"The current reported heading for this launch event is: "
                    (CADDR (CADAR MESSAGE))
                    T T
"HEADING values are directly related to compass values."
                    T "360 degrees is EAST" T
                    "90 degrees is NORTH"
```

```
                    T "180 degrees is WEST" T
                    "270 degrees is SOUTH."
                    T T
"Type in the new value in degrees followed by a carriage re
turn."
                    T
"Type NIL in response to this or to one of the next questio
ns to have SENTINEL evaluate the modified message."
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                T T T)
                    (METASCHEDULER MESSAGE]
          (RPLACA (CDDR (CADAR MESSAGE))
                  READ)
          (printout T T
"The reported BSS sites that should detect this launch even
t are: "
                    (CADDAR MESSAGE)
                    T T
                  "BSS sites are numbered 1 through 7."
                    T
"More than one number for this parameter indicates that mor
e than one BSS site will be able to detect this launch."
                    T T
"Type in the new value for the BSS sites or type NIL to hav
e SENTINEL evaluate the modified message."
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                T T T)
                    (METASCHEDULER MESSAGE]
          (RPLACA (CDDAR MESSAGE)
                  READ)
          (printout T T
"The current value for the SENSOR SITE CONFIDENCE is: "
                    (CADDR MESSAGE)
                    T
"The SITE CONFIDENCE can vary from 0 (LOW) to 9 (HIGH)"
                    T
"type in the new value followed by a carriage return."
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
```

```
           (SETQ READ (RATOM NIL NIL))
           [COND
             ((NULL READ)
               (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                    T T T)
                      (METASCHEDULER MESSAGE]
           (RPLACA (CDDR MESSAGE)
                  READ)
           [COND
             ((EQ LTRSENSOR (QUOTE S2))
               (SETQ SENSOR (QUOTE ADSSOUTH]
           [COND
             ((EQ LTRSENSOR (QUOTE N2))
               (SETQ SENSOR (QUOTE ADSNORTH]
           (printout T T "Current Capability of sensor: "
                     SENSOR T "is reported as: "
                     (GETPROP SENSOR (QUOTE CAPABILITY))
                     T T
"SENSOR CAPABILITY can be FULL, PARTIAL or ZERO. Type in on
e of these three values followed by a carriage return."
                     T)
           (SETQ READ (RATOM NIL NIL))
           [COND
             ((NULL READ)
               (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                    T T T)
                      (METASCHEDULER MESSAGE]
           (PUTPROP SENSOR (QUOTE CAPABILITY)
                   READ)
           (printout T T "Current INTEL CONDITION is: "
                     (GetValue IntelBoard (QUOTE
                                       INTELCONDITION))
                     T T
      "INTEL Conditions fall into these categories: "
                     T "RED: indicate imminent attack." T
                    "YELLOW; indicate possible attack."
                     T
          "GREEN: no attack indications are present."
                     T
       "TEST: indications of a test launch received."
                     T T
"type in the new condition followed by a carriage return."
                     T)
           (SETQ READ (RATOM NIL NIL))
           [COND
             ((NULL READ)
               (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
```

```
                         (METASCHEDULER MESSAGE]
          (PutValue IntelBoard (QUOTE INTELCONDITION)
                    READ)
          (printout T T "INTEL Source is reported to be: "
                    (GetValue IntelBoard (QUOTE SOURCE))
                    T
"INTEL Sources can be OPEN (meaning publicly known or annou
nced) or INTEL (obtained from a classified source)."
                    T "Type in the new value." T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                T T T)
                         (METASCHEDULER MESSAGE]
          (PutValue IntelBoard (QUOTE SOURCE)
                    READ)
          (printout T T
"TIME at which the BSS report for this event is expected is
: "
                    (CAADR MESSAGE)
                    T T
"TIME is a 6 digit number, where the first two digits are t
he hours."
                    T "The next two are minutes." T
                    "And the last two are seconds."
                    T T
               "Type in the new value for this TIME: "
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                T T T)
                         (METASCHEDULER MESSAGE]
          (RPLACA (CADR MESSAGE)
                  READ)
          (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                    T)
          (METASCHEDULER MESSAGE])
(MODIFY2)
51←
NIL
51←
NIL
51←(REDO 100)
```

```
(MODIFY3
  [LAMBDA (MESSAGE)                              (* edited: "20-Sep-85 14:36")
    (PROG (SENSOR READ)

              (* MODIFY3 permits the modification of BSS messages as MODIFY a
nd MODIFY2 permit for ADS-1 and ADS-2 messages. The user can modify various
parameters of the BSS messaged and observe the result when (METASCHEDULER
MESSAGE) is invoked again. Due to its similarity to MODIFY2 and the large numbe
r of PRINTOUT statements, the MODIFY3 code is fairly self documenting.)


                                                   (* What MODIFY2 had to restore
a perfect match, we must do similar for BSS)
          [COND
            ((NULL EVENTMATCH)
              (RETURN (printout T
                                   "There is NO EVENTMATCH:"
                                                T
              "SEQUENCE of messages is out of order or"
                                                T
                      "IF you still wish to use MODIFY,"
                                                T
                "SETQ EVENTMATCH to an existing EVENT."
                                                T]
          (printout T T "BSS REPORT labeled as "
                      EVENTMATCH " can now be modified." T T
            "BSS site which detected this event was: "
                      (CAAADR (CADAR MESSAGE))
                      T
               "BSS sites are numbered 1 through 7. "
                      T T
      "If you wish to change the detecting BSS site, "
                      T
"Type in the new value for the Detecting BSS site number, o
r type NIL to exit the MODIFY program."
                      T)
          (SETQ READ (RATOM NIL NIL))
          (COND
            ((NULL READ)
              (RETURN)))                (* P2TIME property restored to E
VENTMATCH so that it will be matched properly when METASCHEDULER invoked
again.)
          (PUTPROP EVENTMATCH (QUOTE P2TIME)
                    (CADAAR (CDADAR MESSAGE)))
          (RPLACA (CAADR (CADAR MESSAGE))
                   READ)
          (printout T T
"The current reported heading for this launch event is: "
                    (CADDR (CADAR MESSAGE))
                    T T
"HEADING values are directly related to compass values."
                    T "360 degrees is EAST" T
```

```
                    "90 degrees is NORTH"
                    T "180 degrees is WEST" T
                    "270 degrees is SOUTH."
                    T T
"Type in the new value in degrees followed by a carriage re
turn."
                    T
"Type NIL in response to this or to one of the next questio
ns to have SENTINEL evaluate the modified message."
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                T T T)
                      (METASCHEDULER MESSAGE]
          (RPLACA (CDDR (CADAR MESSAGE))
                  READ)
          (printout T T
"The earliest expected impact time for any of the objects d
etected in this event is: "
                    (CAADR MESSAGE)
                    T T
           "SIMCOPE TIME is expressed in 6 digits: "
                    T
       "2 digits each for hours, minutes and seconds."
                    T T
"Type in the new TIME for expected impact or type NIL to ha
ve SENTINEL evaluate the MESSAGE."
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                T T T)
                      (METASCHEDULER MESSAGE]
          (RPLACA (CADR MESSAGE)
                  READ)
          (printout T T
  "The current value for the BSS SITE CONFIDENCE is: "
                    (CADDR MESSAGE)
                    T
"The SITE CONFIDENCE can vary from 0 (LOW) to 9 (HIGH)"
                    T
"type in the new value followed by a carriage return."
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
```

```
              ((NULL READ)
                (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                     T T T)
                        (METASCHEDULER MESSAGE]
          (RPLACA (CDDR MESSAGE)
                  READ)
          [SETQ SENSOR (MKATOM (CONCAT
                                 (QUOTE TGT0)
                                 (CAAADR (CADAR MESSAGE]
          (printout T T "Current Capability of sensor: "
                     (CONVERT SENSOR)
                     T "is reported as: "
                     (GETPROP SENSOR (QUOTE CAPABILITY))
                     T T
"SENSOR CAPABILITY can be FULL, PARTIAL or ZERO. Type in on
e of these three values followed by a carriage return."
                     T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                     T T T)
                        (METASCHEDULER MESSAGE]
          (PUTPROP SENSOR (QUOTE CAPABILITY)
                   READ)
          (printout T T
"The trajectory stability and number of objects detected ca
n now be modified."
                     T T
                    "Current Trajectory stability is: "
                     (CADDAR MESSAGE)
                     T
"where Y indicates a stable ballistic trajectory, and"
                     T
          "N indicates NO, the missile is manuvering."
                     T T
"Type in the new value for trajectory stability or NIL to r
eturn to BSS processing."
                     T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                     T T T)
                        (METASCHEDULER MESSAGE]
          (RPLACA (CDDAR MESSAGE)
                  READ)
```

```
                (METASCHEDULER MESSAGE]
        (RPLACA (CDDAR MESSAGE)
                READ)
        (printout T T
"The number of objects detected can vary from 0 to 5: "
                 T
         "The current number of objects detected is: "
                 (CADADR (CADAR MESSAGE))
                 T T
 "Type in the new value for number of objects or NIL."
                 T)
        (SETQ READ (RATOM NIL NIL))
        [COND
          ((NULL READ)
            (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                             T T T)
                (METASCHEDULER MESSAGE]
        (RPLACA (CDADR (CADAR MESSAGE))
                READ)
        (printout T T
    "TIME at which this BSS message was reported is: "
                 (CADAAR (CDADAR MESSAGE))
                 T T
"TIME is a 6 digit number, where the first two digits are t
he hours."
                 T "The next two are minutes." T
                 "And the last two are seconds."
                 T T
            "Type in the new value for this TIME: "
                 T)
        (SETQ READ (RATOM NIL NIL))
        [COND
          ((NULL READ)
            (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                             T T T)
                (METASCHEDULER MESSAGE]
        (RPLACA (CDAAR (CDADAR MESSAGE))
                READ)
        (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                 T)
        (METASCHEDULER MESSAGE])
(MODIFY3)
64←
NIL
64←
NIL
64←(REDO 100)
```

```
                  (METASCHEDULER MESSAGE]
          (RPLACA (CDDAR MESSAGE)
                  READ)
          (printout T T
"The number of objects detected can vary from 0 to 5: "
                    T
         "The current number of objects detected is: "
                    (CADADR (CADAR MESSAGE))
                    T T
 "Type in the new value for number of objects or NIL."
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                T T T)
                  (METASCHEDULER MESSAGE]
          (RPLACA (CDADR (CADAR MESSAGE))
                  READ)
          (printout T T
    "TIME at which this BSS message was reported is: "
                    (CADAAR (CDADAR MESSAGE))
                    T T
"TIME is a 6 digit number, where the first two digits are t
he hours."
                    T "The next two are minutes." T
                    "And the last two are seconds."
                    T T
              "Type in the new value for this TIME: "
                    T)
          (SETQ READ (RATOM NIL NIL))
          [COND
            ((NULL READ)
              (RETURN (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                                T T T)
                  (METASCHEDULER MESSAGE]
          (RPLACA (CDAAR (CDADAR MESSAGE))
                  READ)
          (printout T T
 "SENTINEL DECISION AID invoked with modified values."
                    T)
          (METASCHEDULER MESSAGE])
(MODIFYS)
64←
NIL
64←
NIL
64←(REDO 100)
```

using $MSGHandler
RuleSet MSGHandler
Last Edited: 3-Oct-85 17:57:15 by

```
WorkSpace Class: Sensors;
Compiler Options:  ;
Temporary Vars:   READ;
Control Structure: DOALL;
********



(* This rule set determines what to do with the message tha
t is passed to it from the METASCHEDULER)


(* The first rule filters out messages that are incorrectly
 addressed)

IF DESTINATION ~= 'CWC
THEN (PRINTOUT T "INCORRECT MESSAGE Addressed to  " DESTINA
TION " Recieved " T "THIS IS THE MESSAGE: " T MESSAGE T
 "IF you wish this message processed TYPE T, otherwise TYPE
 NIL. " T) (SETQ READVAR (RATOM NIL NIL));

(* The second rule will exit out of the Ruleset)

IF READVAR = 'NIL
THEN (SETQ READVAR 'T) (STOP);

(* PRESENTIME is pulled off the ADS1 and ADS2 reports)

IF SENSORTYPE = 'ADS
THEN (SETQ PRESENTIME (CADAAR (CDADAR MESSAGE)));

(* The third rule processes the ADS1 reports, and begins by
 creating the EVENTCOUNT property list)

IF SENSORTYPE = 'ADS
   (GLC SENSORNUMBER) = 1
THEN (SETQ EVENTCOUNT (CONCAT 'EVENT EVCOUNTER))
     (SETQ EVENTCOUNT (MKATOM EVENTCOUNT))
     (SETQ P1TIME (CAADR MESSAGE))
     (PUTPROP EVENTCOUNT 'P1TIME P1TIME)
     (SETQ EVCOUNTER (PLUS EVCOUNTER 1))
     (SCHEDULER MESSAGE);

IF SENSORTYPE = 'ADS
   (GLC SENSORNUMBER) = 2
THEN (ADSCHANDLER MESSAGE)
     (TIMECHECK);
```

```
     (SCHEDULER MESSAGE);

IF SENSORTYPE = 'ADS
   (GLC SENSORNUMBER) = 2
THEN (ADS2HANDLER MESSAGE)
     (TIMECHECK);

IF SENSORTYPE = 'SYS
THEN (SYSHANDLER MESSAGE);

IF SENSORTYPE = 'INT
THEN (INTELHANDLER MESSAGE);

IF SENSORTYPE = 'BSS
THEN (SETQ PRESENTIME (CADAAR (CDADAR MESSAGE)))
     (BSSHANDLER MESSAGE);
#&(RuleSetSource "LQU0.zI[:.YA:.49")
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←
NIL
83←(REDO 100)
```

```
(POLAR
  [LAMBDA (TGTCOORD LNCHCOORD)          (* edited: "23-Oct-85 14:49")

                    (* Given launchsite and target coordinates in terms of latitude and lo
ngitude, POLAR gives the HEADING (90 degrees=North) and RANGE (111 km per
degree, equatorial approximation) from the launchsite to the target. INPUTS "(targ
et coordinates) (launch coordinates)" * OUTPUTS
                    "(HEADING in degrees) (RANGE in km) Decimal fractions are trunca
ted by the FIX command.")


    NIL
    (PROG (ALPHA TOP BOTTOM HYPO)
                                        (* Separates out the latitude and
longitude)
          (SETQ TGTLAT (CAR TGTCOORD))
          (SETQ TGTLONG (CADR TGTCOORD))
          (SETQ LNCHLAT (CAR LNCHCOORD))
          (SETQ LNCHLONG (CADR LNCHCOORD))
                                        (* Sets up the right hand triangle.
 We are looking for alpha which is the arctan of TOP over BOTTOM)
          (SETQ TOP (DIFFERENCE TGTLAT LNCHLAT))
          (SETQ BOTTOM (DIFFERENCE TGTLONG LNCHLONG))
                                        (* SPECIAL CASE; POLAR headi
ngs of 270 (South) and 90 (North) handled separately since BOTTOM=Zero (longit
ude is the same))
          (COND
            ((ZEROP BOTTOM)
              [PROG NIL
                    [COND
                      ((GREATERP TGTLAT LNCHLAT)
                        (SETQ HEADING (QUOTE 270)))
                      (T (SETQ HEADING (QUOTE 90]
                    (SETQ RANGE (TIMES (ABS TOP)
                                       (QUOTE 111]
              (SETQ RANGE (FIX RANGE))
                                        (* THIS TRUNCATES THE DECIM
AL PART OF THE RANGE INFO)
              (printout T "HEADING IS " HEADING
                        "  DEGREES"
                        T "RANGE IS " RANGE "  KILOMETERS"
                        T)
              (RETURN)))                (* SPECIAL CASE; headings of 1
80 (WEST) and 360 (EAST) where TOP=zero)
          (COND
            ((ZEROP TOP)
              [PROG NIL
                    [COND
                      ((GREATERP TGTLONG LNCHLONG)
                        (SETQ HEADING (QUOTE 180)))
                      (T (SETQ HEADING (QUOTE 360]
```

```
                (SETQ HEADING (QUOTE 180)))
              (T (SETQ HEADING (QUOTE 360]
            (SETQ RANGE (TIMES (ABS BOTTOM)
                               (QUOTE 111]
      (SETQ RANGE (FIX RANGE))
      (printout T "HEADING IS " HEADING
                "  DEGREES"
                T "RANGE IS   " RANGE " KILOMETERS"
                T)
      (RETURN)))

                (* We now calculate ALPHA and from there, calculate the hypoteneu
se of the triangle by using the cos, 111 km per degree is what a degree of longitu
de is at the equator.)


      [SETQ ALPHA (ABS (ARCTAN (FQUOTIENT TOP BOTTOM]
      [SETQ RANGE (TIMES (QUOTE 111)
                         (ABS (FQUOTIENT BOTTOM
                                         (COS ALPHA]
                                      (* HEADING correction for II qua
drant)
      [COND
        ((MINUSP TOP)
          (SETQ HEADING (DIFFERENCE (QUOTE 180)
                                    ALPHA]
                                      (* HEADING correction for IV qua
drant)
      [COND
        ((MINUSP BOTTOM)
          (SETQ HEADING (DIFFERENCE (QUOTE 360)
                                    ALPHA]
                                      (* HEADING correction for III qua
drant; form is clumsy due to not having the PLUSP command in Interlisp)
      [COND
        ((AND (NOT (MINUSP TOP))
              (NOT (MINUSP BOTTOM)))
          (SETQ HEADING (PLUS (QUOTE 180)
                              ALPHA]
                                      (* No correction to heading requi
red for I quadrant)
      (COND
        ((AND (MINUSP TOP)
              (MINUSP BOTTOM))
          (SETQ HEADING ALPHA)))
                                      (* Decimal fractions are truncate
d)
      (SETQ HEADING (FIX HEADING))
      (SETQ RANGE (FIX RANGE))
      (printout T "HEADING IS " HEADING " DEGREES" T
                "RANGE IS    "
                RANGE "  KILOMETERS" T T])
```

```
4←(* Regions FPRules)
Using $Regions
RuleSet Regions
Last Edited: 3-Oct-85 17:59:17 by
WorkSpace Class: Launchsite;
Compiler Options: ;
Temporary Vars: ;
Control Structure: DO1;
********



(* This rule set determines what region in the country of T
HEM the launchsite is located in.)

IF LNCHLAT <= 26
   LNCHLONG <= 17
THEN (SETQ REGION 'OUTT);

IF LNCHLAT <= 26
   LNCHLONG > 17
THEN (SETQ REGION 'INN);

IF LNCHLAT > 26
   LNCHLONG >= 18
   LNCHLAT < 33
THEN (SETQ REGION 'WEST);

IF LNCHLAT >= 33
   LNCHLONG > 19
THEN (SETQ REGION 'WEST);

IF LNCHLAT >= 33
   LNCHLONG <= 19
THEN (SETQ REGION 'SOUTH);

(* If launchsite does not fall in the above regions, it is
assumed to be located in the CENTRAL region)

THEN (SETQ REGION 'CENTRAL);
#&(RuleSetSource "LZU0.zI[:.AK9.4")
5←
NIL
5←
NIL
5←
NIL
5←
NIL
5←(REDO 1)
```

```
(SCHEDULER
  [LAMBDA (ADS1REPORT)                    (* edited: "13-Sep-85 13:38")
    (PROG (INFO MATCHLIST MATCHPT MATCHLONG READ)
          (SETQ INFO (CDAR ADS1REPORT))
          (SETQ LNCHLAT (ABS (CADDAR INFO)))
          (SETQ LNCHLONG (CADR INFO))
          (SETQ MATCHLIST LNCHSITELIST)
          (SETQ RESULT NIL)
          [COND
            ((EQ (GNC SENSORNUMBER)
                 (QUOTE S))
              (COND
                ((LESSP LNCHLAT (QUOTE 27.5))
                  (printout T T
"ADS1 REPORT RECEIVED THAT IS OUTSIDE
NORMAL SATELLITE COVERAGE RANGE."
                            T T
      "ADSSOUTH REPORTS LAUNCH EVENT AT LATITUDE OF: "
                            LNCHLAT " degrees." T T
"Normal ADSSOUTH satellite coverage extends from
27.5 to 40 degrees south LATITUDE."
                            T T
            "You are now in a separate break window."
                            T T
"Type GO followed by a carriage return if you want
SENTINEL to process this report"
                            T T
"Otherwise type in ↑: (up arrow: it is the shifted key to
the lower right of the carriage return)
followed by a carriage return
to stop further processing of this report."
                            T T T)
                  (BREAK1 NIL T]
          [COND
            ((EQ (GNC SENSORNUMBER)
                 (QUOTE N))
              (COND
                ((GREATERP LNCHLAT (QUOTE 29.5))
                  (printout T T
"ADS1 REPORT RECEIVED THAT IS OUTSIDE
NORMAL SATELLITE COVERAGE RANGE."
                            T T
      "ADSNORTH REPORTS LAUNCH EVENT AT LATITUDE OF: "
                            LNCHLAT " degrees." T T
"Normal ADSNORTH satellite coverage extends from
15 to 29.5 degrees south LATITUDE."
                            T T
            "You are now in a separate break window."
                            T T
```

```
"Type GO followed by a carriage return if you want
SENTINEL to process this report"
                              T T
"Otherwise type in ↑: (up arrow: it is the shifted key to
the lower right of the carriage return)
followed by a carriage return
to stop further processing of this report."
                              T T T)
                    (BREAK1 NIL T]
      LOOP(COND
            ((NULL (CAR MATCHLIST))
              (RETURN)))
          (SETQ MATCHPT (CAR MATCHLIST))
          (SETQ MATCHLAT (GETPROP MATCHPT (QUOTE LAT)))
          (SETQ MATCHLONG (GETPROP MATCHPT (QUOTE LONG)))

          (* Attempts to match LAT and LONG of all known launchsites to the
LNCHLAT and LNCHLONG announced in the ADS1 Report; matching is done to wit
hin .6 of a degree because of slight sensor variations in reporting launch coordinat
es)


          [COND
            ((LESSP (ABS (DIFFERENCE MATCHLAT LNCHLAT))
                    (QUOTE .6))
              (COND
                ((LESSP (ABS (DIFFERENCE LNCHLONG
                                             MATCHLONG))
                        (QUOTE .6))
                  (SETQ RESULT MATCHPT]
          (SETQ MATCHLIST (CDR MATCHLIST))
          (GO LOOP))
    (printout T "SITE MATCHES " RESULT T T)
    (COND
      ((NULL RESULT)
        (PROG NIL                    (* New launchsites are named LS
15, LS16 and so on as the LSCOUNTER counts up)
              (SETQ LSCOUNTER (PLUS LSCOUNTER 1))
              (SETQ RESULT (CONCAT (QUOTE LS)
                                   LSCOUNTER))
              (SETQ RESULT (MKATOM RESULT))
              (SETQ LNCHSITELIST (APPEND LNCHSITELIST
                                         (LIST RESULT)))
              (PUTPROP RESULT (QUOTE LAT)
                       LNCHLAT)
              (PUTPROP RESULT (QUOTE LONG)
                       LNCHLONG))
        (HIVALCALL RESULT)))         (* BSSCALL and FAN are invoke
d only if the property list of the launchsite does not already contain BSSTAG and F
ANCALL)
    (PUTPROP EVENTCOUNT (QUOTE P1LS)
```

```
     RESULT)
    (COND
      ((NULL (GETPROP RESULT (QUOTE BSSTAG)))
        (BSSCALL RESULT)))          (* THIS IS WITH THE "DUMB" I
MPLEMENTATION OF FANCALL)
    (SETQ TARGETLIST (REVERSE (APPEND TGTAREA4 TGTAREA3
                                      TGTAREA2 TGTAREA1)))
    (COND
      ((NULL (GETPROP RESULT (QUOTE FANCALL)))
        (FAN RESULT)
        (PUTPROP RESULT (QUOTE FANCALL)
                 FANLIST)))

             (* EVENTS are only added to the event list when they are unique, T
his prevents MODIFY from adding many events to the EVENTLIST when only one e
vent is involved)


    [COND
      ((NULL (FMEMB EVENTCOUNT EVENTLIST))
        (SETQ EVENTLIST (APPEND EVENTLIST (LIST EVENTCOUNT]
                                          (* The ADS1 Report can now be
processed by running rule sets)
    (SETQ P1SENSORCONF (CADDR ADS1REPORT))
    (SETQ P1IRINTENSITY (CADAR (CDADAR ADS1REPORT)))
    (RunRS (QUOTE ADS1Confidence))
    (RunRS (QUOTE Regions))          (* Now the user receives the notif
ication and result of processing the ADS1 Report)
    (CF)
    (printout T "ADS1 INDICATION OF LAUNCH RECEIVED:" T T
              "The EVENT NUMBER assigned is: "
              EVENTCOUNT T T "1. DETECTED: " PRESENTIME T
              "2. ADS: "
              (GETPROP EVENTCOUNT (QUOTE P1SENSORTYPE))
              T "3. REGION: " REGION T "4. SITE: "
              (GETPROP EVENTCOUNT (QUOTE P1LSTYPE))
              T "5. LAUNCHER TYPE: " (GETPROP
                EVENTCOUNT
                (QUOTE P1IRTYPE))
              T "6. CONFIDENCE: " (GETPROP EVENTCOUNT
                                           (QUOTE
                                          P1CONFIDENCE))
              T T
"The ADS2 Report for this launch event is expected at: "
              (GETPROP EVENTCOUNT (QUOTE P1TIME))
              " HOURS." T T
  "Further explanation can be obtained by typing a T,"
              T "otherwise type NIL to proceed." T)
                                          (* An rudimentary explanation ca
pability is provided by SCHEDULER which consists of the outlining of all five factor
s which compose the ADS1 confidence level)
    (SETQ READ (RATOM NIL NIL))
```

```
    Further explanation can be obtained by typing a T,
                  T "otherwise type NIL to proceed." T)
                                         (* An rudimentary explanation ca
pability is provided by SCHEDULER which consists of the outlining of all five factor
s which compose the ADS1 confidence level)
    (SETQ READ (RATOM NIL NIL))
    (COND
      ((NULL READ)
        (RunRS (QUOTE ADS1FeatureSet)))
      (T (printout T T "SENSOR confidence is: "
                   (GETPROP EVENTCOUNT (QUOTE P1SENSORCONF)
                           )
                   T "SENSOR capability is: "
                   (GETPROP EVENTCOUNT (QUOTE P1SENSORCAP))
                   T "IR readings classify event as: "
                   (GETPROP EVENTCOUNT (QUOTE P1IRTYPE))
                   T "REPORTED type of launchsite is: "
                   (GETPROP RESULT (QUOTE TYPE))
                   T "Current INTEL condition is: "
                   (GetValue IntelBoard (QUOTE
                                    INTELCONDITION))
                   T "as provided by "
                   (GetValue IntelBoard (QUOTE SOURCE))
                   " sources." T)
                                         (* Finally, to reduce the event list
 down to a feature set, ADS1FeatureSet is invoked)
         (RunRS (QUOTE ADS1FeatureSet])
(SCHEDULER)
17←
NIL
17←
NIL
17←
NIL
17←
NIL
17←
NIL
17←
NIL
17←
NIL
17←
NIL
17←
NIL
17←
NIL
17←
NIL
17←(REDO 100)
```

```
using $SensorCheck
```

RuleSet SensorCheck

```
Last Edited: 9-Oct-85 14:20:29 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   XVAR YVAR FNUMBER SNUMBER SUMNUMBER;
Control Structure: DOALL;
********


(* SensorCheck is invoked by ADS2HANDLER upon a successful
 matching of an ADS1 to an ADS2 event. Using the
  LAUNCHSITE, OUTAGEREASON and RPTSENSORCAP variables of
  the MATCHBoard placed by ADS2HANDLER at the match:
  SensorCheck analyzes this data to provide a COMMENT
  and EXPNUMBER which is then added to MATCHBoard.)

THEN XVAR←MATCHBoard:P2LS
     YVAR←(GETPROP XVAR 'TYPE)
    SNUMBER←4;

(* The default value for SNUMBER is 4)
IF MATCHBoard:OUTAGEREASON = 'TEST
THEN SNUMBER←1;

 (* Wish to err on side of caution, TEST conditions for US
may lead to wrong indications. Other OUTAGE REASONS are map
ped below.) IF MATCHBoard:OUTAGEREASON = 'SABOTAGE
THEN SNUMBER←2;

 IF MATCHBoard:OUTAGEREASON = 'FAILURE
THEN SNUMBER←3;

IF MATCHBoard:OUTAGEREASON = 'MAINTENANCE
THEN SNUMBER←3;

IF MATCHBoard:OUTAGEREASON = 'UNKNOWN
THEN SNUMBER←2;

 (* No info on launchsite means it is unknown) IF YVAR = 'N
IL THEN MATCHBoard:LAUNCHSITE←'UNKNOWN
FNUMBER←2;

(* Known sites are RED RED1 RED2 suspect ones GREEN) IF (GN
C YVAR) = 'R THEN MATCHBoard:LAUNCHSITE←'KNOWN FNUMBER←4;

 IF (GNC YVAR) = 'G THEN MATCHBoard:LAUNCHSITE←'SUSPECTED F
NUMBER←3;
```

```
(* First combine type of site with outage reason) THEN (Run
RS 'Combinatorial) SNUMBER←SUMNUMBER;

 IF MATCHBoard:RPTSENSORCAP = 'FULL THEN FNUMBER←4;

 IF MATCHBoard:RPTSENSORCAP = 'PARTIAL
THEN FNUMBER←3;

 IF MATCHBoard:RPTSENSORCAP = 'ZERO
THEN FNUMBER←2;

(* Then combine result of first with sensor capability)
THEN (RunRS 'Combinatorial)
      MATCHBoard:EXPNUMBER←SUMNUMBER;

 (* Now interpret results only way to get one is below.)

IF MATCHBoard:RPTSENSORCAP = 'ZERO
   MATCHBoard:LAUNCHSITE    = 'UNKNOWN
   MATCHBoard:OUTAGEREASON ~= 'SABOTAGE
THEN MATCHBoard:EXPNUMBER←1;

(* The COMMENTS are now attached to MATCHBoard)

IF MATCHBoard:EXPNUMBER = 4
THEN MATCHBoard:COMMENT←'(PRINTOUT T "Credibility of Sensor
: VERY GOOD " T);

IF MATCHBoard:EXPNUMBER = 3
THEN MATCHBoard:COMMENT←'(PRINTOUT T "Credibility of Sensor
: GOOD " T);

IF MATCHBoard:EXPNUMBER = 2
THEN MATCHBoard:COMMENT←'(PRINTOUT T "Credibility of Sensor
: POOR " T "Due to: " CAUSE T );

IF MATCHBoard:EXPNUMBER = 1
THEN MATCHBoard:COMMENT←'(PRINTOUT T "Credibility of Sensor
: VERY POOR " T "Due to: " CAUSE T );

THEN (PRINTOUT T T  "This is launchsite type: " MATCHBoard:
LAUNCHSITE T "This is sensor capability: " MATCHBoard:RPTSE
NSORCAP T "This is outage reason: " MATCHBoard:OUTAGEREASON
) (PRINTOUT T "REPORTING SENSOR CHECK reveals" T) (EVAL (Ge
tValue MATCHBoard 'COMMENT));

(* Note this really is not part of the comment but can be d
uplicated easily) IF MATCHBoard:EXPNUMBER <= 2
THEN (PRINTOUT T (GetValue MATCHBoard 'OUTAGEREASON) T);
```

```
(* Note this really is not part of the comment but can be d
uplicated easily) IF MATCHBoard:EXPNUMBER <= 2
THEN (PRINTOUT T (GetValue MATCHBoard 'OUTAGEREASON) T);

THEN (SETQ P1IRTYPE (GETPROP EVENTMATCH 'P1IRTYPE))
 MATCHBoard:LAUNCHERTYPE←'TYPE2;

 IF P1IRTYPE = 'TYPE1 P2IRTYPE = 1 THEN MATCHBoard:LAUNCHER
TYPE←'TYPE1;
#&(RuleSetSource "GWU0.zI[:.PG8.35")
19←
NIL
19←
NIL
19←
NIL
19←
NIL
19←
NIL
19←(PP SYSHANDLER)
(SYSHANDLER
  [LAMBDA (SYSMSG)                          (* edited: "22-Jul-85 16:42")
    (PROG (OUTAGETYPE SENSORCAPABILITY TIMETORESTORE
                      OUTAGETIME SENSORTYPE SUBSYSTEM
                      FRAGMENT REASON SYSTEMSG)
          (SETQ SYSTEMSG SYSMSG)
          (SETQ SENSORCAPABILITY (CADR SYSTEMSG))
          (SETQ TIMETORESTORE (CADDAR SYSTEMSG))
          (SETQ FRAGMENT (CDADAR SYSTEMSG))
          (SETQ OUTAGETYPE (CAAR FRAGMENT))
          (SETQ OUTAGETIME (CADR FRAGMENT))
          (SETQ SENSORTYPE (CADAR FRAGMENT))
          (SETQ SUBSYSTEM (CADDAR FRAGMENT))
          (SETQ REASON (CADR (CDDAR FRAGMENT)))
          (printout T "THIS IS OUTAGE TYPE " OUTAGETYPE T
                    "THIS IS SENSORCAPABILITY "
                    SENSORCAPABILITY T
                    "THIS IS TIMETORESTORE "
                    TIMETORESTORE T "THIS IS OUTAGETIME "
                    OUTAGETIME T "THIS IS SENSORTYPE "
                    SENSORTYPE T "THIS IS SUBSYSTEM  "
                    SUBSYSTEM T "THIS IS REASON " REASON T)
          (RunRS (QUOTE SYSAnalyst])
(SYSHANDLER)
20←
NIL
20←
```

using $sensormap

RuleSet SensorMap

Last Edited: 3-Oct-85 18:37:44 by

```
WorkSpace Class: Sensors;
Compiler Options:  ;
Temporary Vars:  ;
Control Structure: DOALL;
********



(* SensorMap permits the numerical sensor rating that is ca
rried in the EVENTLIST to be broken back down into its cons
tituent parts: SENSORCONF and SENSORCAPABILITY.)

IF SENSORNUM = 9
THEN (SETQ SENSORCAP 'FULL)
     (SETQ SENSORCONF 'HIGH);

IF SENSORNUM = 8
THEN (SETQ SENSORCAP 'FULL)
     (SETQ SENSORCONF 'MEDUIM);

IF SENSORNUM = 7
THEN (SETQ SENSORCAP 'PARTIAL)
     (SETQ SENSORCONF 'HIGH);

IF SENSORNUM = 6
THEN (SETQ SENSORCAP 'FULL)
     (SETQ SENSORCONF 'LOW);

IF SENSORNUM = 5
THEN (SETQ SENSORCAP 'PARTIAL)
     (SETQ SENSORCONF 'MEDUIM);

IF SENSORNUM = 4
THEN (SETQ SENSORCAP 'ZERO)
     (SETQ SENSORCONF 'HIGH);

IF SENSORNUM = 3
THEN (SETQ SENSORCAP 'PARTIAL)
     (SETQ SENSORCONF 'LOW);

IF SENSORNUM = 2
THEN (SETQ SENSORCAP 'ZERO)
     (SETQ SENSORCONF 'MEDUIM);

IF SENSORNUM = 1
THEN (SETQ SENSORCAP 'ZERO)
     (SETQ SENSORCONF 'LOW);
```

```
(SENTINEL
  [LAMBDA NIL                                    (* edited: " 8-Oct-85 14:40")
                                                 (* This function restores the lists,
 counter and LOOPS objects used in SENTINEL to their original condition.)
    (PROG (MATCHLIST MATCHPOINT MLIST MPOINT)
          (SETQ READVAR (QUOTE T))
          (SETQ LNCHLONG NIL)
          (SETQ LNCHLAT NIL)
          (SETQ LSCOUNTER (QUOTE 14))
          (SETQ INTCOUNTER 0)
          (SETQ CAUSE (QUOTE UNCLEAR))
                                                 (* This part removes the property
 lists of any launchsites added to the LNCHSITELIST after LS14; that is after any pr
eviously unknown launchsite)
          (PROG (ZAPLIST ZAPPOINT)
                (SETQ ZAPLIST (FMEMB (QUOTE LS15)
                                     LNCHSITELIST))
            LOOP(COND
                  ((NULL (CAR ZAPLIST))
                    (RETURN)))
                (SETQ ZAPPOINT (CAR ZAPLIST))
                (REMPROP ZAPPOINT (QUOTE LAT))
                (REMPROP ZAPPOINT (QUOTE LONG))
                (REMPROP ZAPPOINT (QUOTE HIVALTAG))
                (REMPROP ZAPPOINT (QUOTE BSSTAG))
                (REMPROP ZAPPOINT (QUOTE FANCALL))
                (SETQ ZAPLIST (CDR ZAPLIST))
                (GO LOOP))                       (* This part removes the property
 list of all past events whose tag is stored in the EVENTLIST; EVENTLIST is restor
ed afterwards)
          (PROG (ZPOINT)
            LOOP(COND
                  ((NULL (CAR EVENTLIST))
                    (RETURN)))
                (SETQ ZPOINT (CAR EVENTLIST))
                (REMPROP ZPOINT (QUOTE P1TIME))
                (REMPROP ZPOINT (QUOTE P1SENSOR))
                (REMPROP ZPOINT (QUOTE P1IRTYPE))
                (REMPROP ZPOINT (QUOTE P1SENSORCONF))
                (REMPROP ZPOINT (QUOTE P1SENSORTYPE))
                (REMPROP ZPOINT (QUOTE P1SENSORCAP))
                (REMPROP ZPOINT (QUOTE P1CONFIDENCE))
                (REMPROP ZPOINT (QUOTE P1LSTYPE))
                (REMPROP ZPOINT (QUOTE P1LS))
                (REMPROP ZPOINT (QUOTE P2TIME))
                (REMPROP ZPOINT (QUOTE P2TYPE))
                (REMPROP ZPOINT (QUOTE P2SENSOR))
                (REMPROP ZPOINT (QUOTE P2CONFIDENCE))
                (REMPROP ZPOINT (QUOTE P2HYPO))
                (REMPROP ZPOINT (QUOTE P2HEADING))
```

```
          (GO LOOP))              (* The target and launchsite lists
are initialized and EVENTLIST is initialized so that events can be CONSed to it)
          (SETQ EVCOUNTER 1)
          (SETQ EVENTLIST (LIST (QUOTE EVENT0)))
          (SETQ EVENTLIST (REMOVE (QUOTE EVENT0)
                                  EVENTLIST))
          (SETQ LNCHSITELIST
            (QUOTE (LS1 LS2 LS3 LS4 LS5 LS6 LS7 LS8 LS9
                    LS10 LS11 LS12 LS13 LS14)))
          (SETQ BSSLIST
            (QUOTE (TGT01 TGT02 TGT03 TGT04 TGT05 TGT06
                     TGT07)))
          (SETQ HIVALTGTLIST
            (QUOTE (TGT1 TGT2 TGT3 TGT4 TGT5 TGT6)))
          (SETQ TGTAREA1
            (QUOTE (TGT7 TGT8 TGT9 TGT10 TGT11 TGT12 TGT13
                     TGT14)))
          (SETQ TGTAREA2
            (QUOTE (TGT15 TGT16 TGT17 TGT18 TGT19 TGT20
                     TGT21 TGT22)))
          (SETQ TGTAREA3
            (QUOTE (TGT23 TGT24 TGT25 TGT26 TGT27)))
          (SETQ TGTAREA4
            (QUOTE (TGT28 TGT29 TGT30 TGT31 TGT32 TGT33)))
                                  (* Insures that the HIVALTAG pro
perty is embedded on the property list of each launchsite)
          (PROG (LSHIVAL HIVALIST)
                (SETQ HIVALIST LNCHSITELIST)
            LOOP(COND
                  ((NULL (CAR HIVALIST))
                    (RETURN)))
                (SETQ LSHIVAL (CAR HIVALIST))
                (COND
                  ((NULL (GETPROP LSHIVAL (QUOTE HIVALTAG))
                         )
                    (HIVALCALL LSHIVAL)))
                (SETQ HIVALIST (CDR HIVALIST))
                (GO LOOP))

          (* This erases the embedded results of any SYS messages on all the
sensors, BSS sites, ADSSOUTH, ADSNORTH and all the C3 network or high value
targets CAPABILITY of all sensors reset to FULL)


          (PROG (ZAPLIST ZAPPOINT)
                (SETQ ZAPLIST (APPEND BSSLIST HIVALTGTLIST)
                  )
                (SETQ ZAPLIST (CONS (QUOTE ADSNORTH)
                                    ZAPLIST))
                (SETQ ZAPLIST (CONS (QUOTE ADSSOUTH)
                                    ZAPLIST))
```

```
                                         ZAPLIST))
            LOOP(COND
                  ((NULL (CAR ZAPLIST))
                    (RETURN)))
                (SETQ ZAPPOINT (CAR ZAPLIST))
                (PUTPROP ZAPPOINT (QUOTE CAPABILITY)
                         (QUOTE FULL))
                (REMPROP ZAPPOINT (QUOTE REASON))
                (REMPROP ZAPPOINT (QUOTE TYPE))
                (REMPROP ZAPPOINT (QUOTE RESTORETIME))
                (REMPROP ZAPPOINT (QUOTE OUTTIME))
                (SETQ ZAPLIST (CDR ZAPLIST))
                (GO LOOP))          (* This clears the IntelBoard and
the SYSBoard of all past values)
          (PutValue IntelBoard (QUOTE SOURCE)
                    NIL)
          (PutValue IntelBoard (QUOTE SOURCECOUNTRY)
                    NIL)
          (PutValue IntelBoard (QUOTE INTELCONDITION)
                    NIL)
          (PutValue IntelBoard (QUOTE INTELTYPE)
                    NIL)
          (PutValue IntelBoard (QUOTE REPORTEDCOUNTRY)
                    NIL)
          (PutValue IntelBoard (QUOTE COMMENT)
                    NIL)
          (PutValue IntelBoard (QUOTE EXPNUMBER)
                    0)
          (PutValue SYSBoard (QUOTE OUTAGEREASON)
                    NIL)
          (PutValue SYSBoard (QUOTE TYPEOFOUTAGE)
                    NIL)
          (PutValue SYSBoard (QUOTE TYPEOFSENSOR)
                    NIL)
          (PutValue SYSBoard (QUOTE RESTORATIONTIME)
                    NIL)
          (PutValue SYSBoard (QUOTE CAPABILITYOFSENSOR)
                    NIL)
          (PutValue SYSBoard (QUOTE OUTSENSOR)
                    0)
          (PutValue SYSBoard (QUOTE OUTTEST)
                    0)
          (PutValue SYSBoard (QUOTE OUTSABOTAGE)
                    0)
          (PutValue SYSBoard (QUOTE AFFSENSOR)
                    0)
          (PutValue SYSBoard (QUOTE OUTFAILURE)
                    0)
          (PutValue SYSBoard (QUOTE OUTTOTAL)
                    0)
```

```
                (PutValue SYSBoard (QUOTE OUTTOTAL)
                          0)
                (PutValue SYSBoard (QUOTE OUTEMERG)
                          0)
                (PutValue SYSBoard (QUOTE COMMENT)
                          (QUOTE (printout T T
                                 "NO SYS Reports Received. "
                                                T
                             "OVERALL SYS STATUS: VERY GOOD"
                                                T)))
                (PutValue SYSBoard (QUOTE EXPNUMBER)
                          4)
                (PutValue SYSBoard (QUOTE AGSNUMBER)
                          0)
                (PutValue SYSBoard (QUOTE BGSNUMBER)
                          0)
                (PutValue SYSBoard (QUOTE AFULL)
                          4)
                (PutValue SYSBoard (QUOTE BFULL)
                          4)
                (PutValue SYSBoard (QUOTE APARTIAL)
                          0)
                (PutValue SYSBoard (QUOTE BPARTIAL)
                          0)
                (PutValue SYSBoard (QUOTE AZERO)
                          0)
                (PutValue SYSBoard (QUOTE BZERO)
                          0)
                (SETQ MESSAGE NIL)
                (PUTPROP (QUOTE ADSSOUTH)
                         (QUOTE NAME)
                         (QUOTE ADSSOUTH))
                (PUTPROP (QUOTE ADSNORTH)
                         (QUOTE NAME)
                         (QUOTE ADSNORTH))
                (SETQ PLACE NIL)
                (printout T
                    "SENTINEL DECISION AID is initialized. "
                          T T)
                (printout T
"If assistance is needed type (HELP), otherwise proceed."
                          T])
(SENTINEL)
35←
NIL
35←
NIL
35←
NIL
35←(REDO 100)
```

using $Speculate
RuleSet Speculate
Last Edited: 7-Oct-85 15:18:16 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   XVAR , YVAR , MARK;
Control Structure: DOALL;
********



(* Speculate analyzes patterns to determine whether certain
 special cases such as TEST, SABOTAGE or FIRSTSTRIKE are pr
esent. It uses PLACE and  puts a COMMENT and an EXPFACT on
EXPLAINBoard to record its results.)

THEN EXPLAINBoard:COMMENT←'NIL
     EXPLAINBoard:EXPFACT←'NIL
     XVAR←0 YVAR←0 (SETQ PLACE 'NIL);

(* First special case tackled is TEST.)

IF CAUSE = 'TEST
THEN XVAR←(PLUS XVAR 1);

IF (GETPROP 'ADSNORTH 'REASON) = 'TEST
THEN XVAR←(PLUS XVAR 1)
     (SETQ PLACE 'ADSNORTH);

IF (GETPROP 'ADSSOUTH 'REASON) = 'TEST
THEN XVAR←(PLUS XVAR 1)
     (SETQ PLACE 'ADSSOUTH);

IF (GETPROP 'TGT2 'REASON) = 'TEST
THEN XVAR←(PLUS XVAR 1)
     (SETQ PLACE 'BRF);

IF (GETPROP 'TGT4 'REASON) = 'TEST
THEN XVAR←(PLUS XVAR 1)
     (SETQ PLACE 'CWC);

IF (GETPROP 'TGT5 'REASON) = 'TEST
THEN XVAR←(PLUS XVAR 1)
     (SETQ PLACE 'ADSGSF);

IF (GETPROP 'TGT6 'REASON) = 'TEST
THEN XVAR←(PLUS XVAR 1)
     (SETQ PLACE 'BSSGSF);

IF PLACE = 'NIL
```

```
IF PLACE = 'NIL
THEN (SETQ PLACE SYSBoard:TYPEOFSENSOR);

IF XVAR >= 1
THEN EXPLAINBoard:EXPFACT←'TEST
     EXPLAINBoard:COMMENT←'(PRINTOUT T T "WARNING: US SENSO
R running TESTS at:"  PLACE T) (STOP);

(* Second case is the FIRSTSTRIKE.)

IF TRENDBoard:HIVALNUMBER = 3 MATCHBoard:EXPNUMBER = 3
THEN EXPLAINBoard:EXPFACT←'FIRSTRIKE
   EXPLAINBoard:COMMENT←'(PRINTOUT T T "SOME FIRST STRIKE I
NDICATIONS ARE PRESENT." T) (STOP);

IF TRENDBoard:HIVALNUMBER > 3 MATCHBoard:EXPNUMBER > 3
THEN EXPLAINBoard:EXPFACT←'FIRSTSTRIKE
     EXPLAINBoard:COMMENT←'(PRINTOUT T T "FIRST STRIKE INDI
CATIONS ARE PRESENT!" T) (STOP);

 (* Third special case is SABOTAGE.)

IF (GetValue SYSBoard 'CAUSE) = 'SABOTAGE
THEN YVAR←(PLUS YVAR 1);

IF MATCHBoard:OUTAGEREASON = 'SABOTAGE
THEN YVAR←(PLUS YVAR 1);

IF (GetValue SYSBoard 'OUTSABOTAGE) >= 2
   (GetValue SYSBoard 'AGSNUMBER) < 4
THEN YVAR←(PLUS YVAR 1);

 IF YVAR >= 2
THEN EXPLAINBoard:EXPFACT←'SABOTAGE
  EXPLAINBoard:COMMENT←'(PRINTOUT T T "WARNING: WIDESPREAD
indications of SABOTAGE present." T) (STOP);

(* Fourth case, P2BSS is compared to SENTINEL caluculations
 and site are checked for FULL capability.)

THEN XVAR←(MKATOM (CONCAT 'TGT0 (GNC P2BSS)))
     YVAR←(MKATOM (CONCAT 'TGT0 (GLC P2BSS)))
     EXPLAINBoard:EXPFACT←'NIL;

IF (EQMEMB XVAR DetectBoard:WILLDETECT) = 'T
   (EQMEMB YVAR DetectBoard:WILLDETECT) = 'T
THEN MARK←'MATCH;

IF MARK = 'MATCH
   (GETPROP XVAR 'CAPABILITY) = 'FULL
```

```
IF (EQMEMB XVAR DetectBoard:WILLDETECT) = 'T
   (EQMEMB YVAR DetectBoard:WILLDETECT) = 'T
THEN MARK←'MATCH;

IF MARK = 'MATCH
   (GETPROP XVAR 'CAPABILITY) = 'FULL
   (GETPROP YVAR 'CAPABILITY) = 'FULL
THEN MARK←'FULLMATCH;

IF MARK = 'MATCH
THEN EXPLAINBoard:EXPFACT←'BSS
  EXPLAINBoard:COMMENT←'(PRINTOUT T T "SENTINEL concurs wit
h ADS-2 BSS prediction." T "However, BSS sites that should
detect are impaired." T);

IF MARK = 'FULLMATCH
THEN EXPLAINBoard:EXPFACT←'BSSFULL
  EXPLAINBoard:COMMENT←'(PRINTOUT T T "SENTINEL concurs wit
h ADS-2 BSS prediction." T "Intercepting BSS sites are in F
ULL operation." T);
#&(RuleSetSource "SIU0.zI[:.Je8.16")
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←
NIL
44←(REDO 100)
```

Using $SYSAnalyst

RuleSet SYSAnalyst

Last Edited: 7-Oct-85 15:05:00 by

```
WorkSpace Class: Reports;
Compiler Options:  ;
Temporary Vars:   OLDCAP OLDTYPE TAG XVAR;
Control Structure: DOALL;
********



(* This ruleset analyzes the incoming SYS reports, and post
s values on the SYSBoard)

THEN XVAR←'NIL OLDCAP←'NIL;

IF OUTAGETYPE = 'E
THEN (SETQ OUTAGETYPE 'EMERGENCY);

IF OUTAGETYPE = 'R
THEN (SETQ OUTAGETYPE 'ROUTINE);

IF SENSORCAPABILITY = 'F
THEN (SETQ SENSORCAPABILITY 'FULL);

IF SENSORCAPABILITY = 'P
THEN (SETQ SENSORCAPABILITY 'PARTIAL);

IF SENSORCAPABILITY = 'Z
THEN (SETQ SENSORCAPABILITY 'ZERO);

IF REASON = 'T
THEN (SETQ REASON 'TEST);

IF REASON = 'M
THEN (SETQ REASON 'MAINTENANCE);

IF REASON = 'F
THEN (SETQ REASON 'FAILURE);

IF REASON = 'S
THEN (SETQ REASON 'SABOTAGE);

IF REASON = 'U
THEN (SETQ REASON 'UNKNOWN);

(* If sensortype=number, then it is a BSS site, embed info
on the property list of the BSS site as well as updating th
e SYSBoard.)
```

```
e SYSBoard.)

IF (NUMBERP SENSORTYPE)
THEN TAG←(MKATOM (CONCAT 'TGT0 SENSORTYPE)) XVAR←'TRUE;

(* Then the Tag is created for the rest of the sensors)

IF SENSORTYPE = 'A
THEN TAG←'TGT5;

IF SENSORTYPE = 'B
THEN TAG←'TGT6;

IF SENSORTYPE = 'C
THEN TAG←'TGT3;

IF SENSORTYPE = 'F
THEN TAG←'TGT2;

IF SENSORTYPE = 'N
THEN  TAG←'ADSNORTH XVAR←'TRUE;

IF SENSORTYPE = 'S
THEN TAG←'ADSSOUTH XVAR←'TRUE;

(* Insures two or more SYS reports from a single sensor are
 not logged as two or more SENSORS that are OUT)

THEN OLDCAP←(GETPROP TAG 'CAPABILITY)
    SYSBoard:OUTTOTAL←(PLUS SYSBoard:OUTTOTAL 1);

IF OLDCAP = 'PARTIAL
THEN SYSBoard:AFFSENSOR←(SUB1 SYSBoard:AFFSENSOR);

IF OLDCAP = 'ZERO
THEN SYSBoard:OUTSENSOR←(SUB1 SYSBoard:OUTSENSOR);

(* The appropriate Instance Variables are updated on the SY
SBoard for all sensor sites, including C3 ones.)

IF SENSORCAPABILITY = 'PARTIAL
THEN SYSBoard:AFFSENSOR←(PLUS SYSBoard:AFFSENSOR 1);

IF REASON = 'FAILURE
THEN SYSBoard:OUTFAILURE←(PLUS SYSBoard:OUTFAILURE 1);

IF SENSORCAPABILITY = 'ZERO
THEN SYSBoard:OUTSENSOR←(PLUS SYSBoard:OUTSENSOR 1);

IF REASON = 'TEST
```

```
IF REASON = 'TEST
THEN SYSBoard:OUTTEST←(PLUS SYSBoard:OUTTEST 1)
     SYSBoard:TYPEOFSENSOR←(GETPROP TAG 'NAME);

IF  REASON = 'SABOTAGE
THEN SYSBoard:OUTSABOTAGE←(PLUS SYSBoard:OUTSABOTAGE 1);

IF OUTAGETYPE = 'EMERGENCY
THEN SYSBoard:OUTEMERG←(PLUS SYSBoard:OUTEMERG 1);

THEN (PUTPROP TAG 'CAPABILITY SENSORCAPABILITY)
     (PUTPROP TAG 'REASON REASON)
     (PUTPROP TAG 'TYPE OUTAGETYPE)
     (PUTPROP TAG 'RESTORETIME TIMETORESTORE)
     (SETQ SENSORTYPE (GETPROP TAG 'NAME)) (CF)
(PRINTOUT T "SYS message Acknowledged." T OUTAGETYPE " OUTA
GE Reported at: " SENSORTYPE T "CAUSE: " REASON T "CURRENT
CAPABILITY OF SENSOR: " SENSORCAPABILITY T "TIME OF OUTAGE:
 " OUTAGETIME T "Restoration expected at: " TIMETORESTORE T
 T T);

(* The SYSBoard is updated to include a Comment)
THEN (RunRS 'SYSComment);

(* C3 node reports are sent to the C3Analyst)

IF XVAR = 'NIL
THEN (RunRS 'C3Analyst);

IF SYSBoard:OUTSENSOR ~= 0
THEN (PRINTOUT T T SYSBoard:OUTSENSOR " out of 9 sensor sit
es are OUT" T "with ZERO CAPABILITY." T T);

IF SYSBoard:AFFSENSOR ~= 0
THEN (PRINTOUT T T SYSBoard:AFFSENSOR " out of 9 sensor sit
es are AFFECTED" T " with PARTIAL CAPABILITY." T T);

IF SYSBoard:OUTTOTAL ~= 0
THEN (PRINTOUT T SYSBoard:OUTTOTAL " SYS reports have been
acknowledged." T "Out of these reports:" T T SYSBoard:OUTEM
ERG " have been EMERGENCY reports." T SYSBoard:OUTSABOTAGE
" have been due to SABOTAGE." T SYSBoard:OUTTEST " have bee
n due to TESTING." T SYSBoard:OUTFAILURE " have been due to
 FAILURE." T);

THEN  (EVAL SYSBoard:COMMENT);
#&(RuleSetSource "LVU0.zI[:.P9B.30")
58←
NIL
58←(REDO 100)
```

```
ôô←(* SYSCheck FPRules)
Using $SYSCheck
RuleSet SYSCheck
Last Edited: 3-Oct-85 18:08:18 by
WorkSpace Class: Sensors;
Compiler Options:  ;
Temporary Vars:   POINTAG TAG MATCHPOINT;
Control Structure: WHILEALL;
Iteration Condition:   T;
********



(* Given a list of C3 sites, SENSORCHECK uses a While LOOP
to see what the CAPABILITY of each site is. One PARTIAL deg
rades the STATUS to PARTIAL, and One ZERO degrades it to ZE
RO. ALL of them must be FULL for STATUS=FULL.)

{1!} THEN (SETQ POINTAG (CAR METALIST))
     (SETQ MATCHLIST (EVAL (CAR METALIST)));

(* The above "one shot bang" rules permits the use of a MET
ALIST that contains all the C3 paths is tested and
executed only once here. Otherwise MATCHLIST is continually
 refreshed and never gets smaller= Infinite LOOP)

THEN (SETQ MATCHPOINT (CAR MATCHLIST))
     (SETQ MATCHLIST  (CDR MATCHLIST));

(* Helps C3Board determine if AGS or BGS link affected)

IF (GNC POINTAG) = 'A
THEN TAG←'AGS;

IF (GNC POINTAG) = 'B
THEN TAG←'BGS;

IF (GETPROP MATCHPOINT 'CAPABILITY) = 'FULL
     STATUS ~= 'PARTIAL
     STATUS ~= 'ZERO
THEN (SETQ STATUS 'FULL);

IF (GETPROP MATCHPOINT 'CAPABILITY) = 'PARTIAL
     STATUS ~= 'ZERO
THEN (SETQ STATUS 'PARTIAL);

IF (GETPROP MATCHPOINT 'CAPABILITY) = 'ZERO
THEN (SETQ STATUS 'ZERO);

(* Once a C3 path list has been exhausted: C3BOARD is calle
```

```
(* Once a C3 path list has been exhausted: C3BOARD is calle
d to update the status number of each path. The next C3 pat
h is checked from METALIST.)

IF (CAR MATCHLIST) = 'NIL
THEN (RunRS 'C3Board)
     (SETQ STATUS 'FULL)
     (SETQ METALIST (CDR METALIST))
     (SETQ MATCHLIST (EVAL (CAR METALIST)))
     (SETQ POINTAG (CAR METALIST));

(* To terminate the WHILEALL loop, a STOP condition is
  activated when all the C3 paths contained in METALIST
  are exhausted)

IF (CAR METALIST) = 'NIL
THEN (STOP);
#&(RuleSetSource "GWU0.zI[:.PG8.33")
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←
NIL
70←(REDO 100)
```

using $SYSComment

RuleSet SYSComment

```
Last Edited: 3-Oct-85 18:06:33 by
WorkSpace Class: Reports;
Compiler Options:  ;
Temporary Vars:   PCT;
Control Structure: DOALL;
********


(* SYSComment examines the pattern of SYS reports recieved
by combining the number of sensors that are OUT and the one
s that are AFFECTED, together with the number of EMERGENCY
reports. These are then set into COMMENT on the SYSBoard al
ong with a cause.)

IF SYSBoard:AFFSENSOR < 2
THEN (SETQ SNUMBER 4);

IF SYSBoard:AFFSENSOR >= 2
   SYSBoard:AFFSENSOR <= 3
THEN (SETQ SNUMBER 3);

IF SYSBoard:AFFSENSOR > 3
   SYSBoard:AFFSENSOR <= 6
THEN (SETQ SNUMBER 2);

IF SYSBoard:AFFSENSOR > 6
THEN (SETQ SNUMBER 1);

IF SYSBoard:OUTSENSOR = 0
THEN (SETQ FNUMBER 4);

IF SYSBoard:OUTSENSOR = 1
THEN (SETQ FNUMBER 3);

IF SYSBoard:OUTSENSOR >= 2
   SYSBoard:OUTSENSOR <= 3
THEN (SETQ FNUMBER 2);

IF SYSBoard:OUTSENSOR > 3
THEN (SETQ FNUMBER 1);

(* This first running of Combinatorial combines the number
of OUT and AFFECTED sensors);

THEN (RunRS 'Combinatorial)
     (SETQ SNUMBER SUMNUMBER) PCT←'NIL;
```

```
(* The percentage of EMERGENCY to TOTAL SYS reports is comp
ared and the FNUMBER set)

THEN PCT←(FQUOTIENT SYSBoard:OUTEMERG SYSBoard:OUTTOTAL);

IF PCT <= .25
THEN (SETQ FNUMBER 4);

IF PCT > .25
   PCT <= .5
THEN (SETQ FNUMBER 3);

IF PCT > .5
   PCT <= .75
THEN (SETQ FNUMBER 2);

IF PCT > .75
THEN (SETQ FNUMBER 1);

(* Default condition when reports < 4 set FNUMBER 3)

IF SYSBoard:OUTTOTAL < 4
THEN (SETQ FNUMBER 3);

(* Second COMBINATORIAL combines sensor OUT numbers as SNUM
BER and Emergency reports as FNUMBER)

THEN (RunRS 'Combinatorial);

(* Cause established, based on which one is greater. Two ti
e results in joint cause. All tie is UNCLEAR cause)

IF SYSBoard:OUTSABOTAGE > SYSBoard:OUTFAILURE
   SYSBoard:OUTSABOTAGE > SYSBoard:OUTTEST
THEN (SETQ CAUSE 'SABOTAGE);

IF SYSBoard:OUTFAILURE > SYSBoard:OUTTEST
   SYSBoard:OUTFAILURE > SYSBoard:OUTSABOTAGE
THEN (SETQ CAUSE 'FAILURE);

IF SYSBoard:OUTTEST > SYSBoard:OUTFAILURE
   SYSBoard:OUTTEST > SYSBoard:OUTSABOTAGE
THEN (SETQ CAUSE 'TEST);

IF SYSBoard:OUTSABOTAGE = SYSBoard:OUTFAILURE
   SYSBoard:OUTSABOTAGE > SYSBoard:OUTTEST
THEN (SETQ CAUSE '"SABOTAGE and FAILURE");

IF SYSBoard:OUTFAILURE = SYSBoard:OUTTEST
```

```
THEN (SETQ CAUSE '"SABOTAGE and FAILURE");

IF SYSBoard:OUTFAILURE = SYSBoard:OUTTEST
   SYSBoard:OUTFAILURE > SYSBoard:OUTSABOTAGE
THEN (SETQ CAUSE '"TEST and FAILURE");

IF SYSBoard:OUTSABOTAGE = SYSBoard:OUTTEST
   SYSBoard:OUTSABOTAGE > SYSBoard:OUTFAILURE
THEN (SETQ CAUSE '"TEST and SABOTAGE");

IF SYSBoard:OUTSABOTAGE = SYSBoard:OUTTEST
   SYSBoard:OUTSABOTAGE = SYSBoard:OUTFAILURE
THEN (SETQ CAUSE 'UNCLEAR);

(* Finally, the SYSBoard is filled with a COMMENT and an EX
PNUMBER)

IF SUMNUMBER = 4
THEN SYSBoard:EXPNUMBER←4
     SYSBoard:COMMENT←'(PRINTOUT T T "Few SYS EMERGENCY Rep
orts Received." T "OVERALL SYS STATUS: VERY GOOD." T);

IF SUMNUMBER = 3
THEN SYSBoard:EXPNUMBER←3
     SYSBoard:COMMENT← '(PRINTOUT T T "Some SYS EMERGENCY R
eports Received." T "OVERALL SYS STATUS: GOOD." T);

IF SUMNUMBER = 2
THEN SYSBoard:EXPNUMBER←2
     SYSBoard:COMMENT← '(PRINTOUT T T "MANY SYS EMERGENCY R
eports Received." T "OVERALL SYS STATUS: POOR." T "Leading
Cause: " CAUSE T);

IF SUMNUMBER = 1
THEN SYSBoard:EXPNUMBER←1
     SYSBoard:COMMENT← '(PRINTOUT T T "VERY MANY SYS EMERGE
NCY REPORTS Received." T "OVERALL SYS STATUS: VERY POOR." T
 "Leading Cause: " CAUSE T);
#&(RuleSetSource "GWU0.zI[:.PG8.34")
79←
NIL
79←
NIL
79←
NIL
79←
NIL
79←
NIL
79←(REDO 100)
```

```
25←(← TargetCheck PPRules)
Using $TargetCheck
RuleSet TargetCheck
Last Edited: 3-Oct-85 18:42:27 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   TGTWILLDETECT TGTCLOSEDETECT TGTFARDETECT
 TLIST TPOINT TRANGE THEADING TARGET THEADIFF MARK TLIST RA
NGEMARK MAXRANGE LSTYPE RANGEDIFF;
Control Structure: WHILEALL;
Iteration Condition:   T;
********



{1!} THEN TLIST←(CDR TAGLIST)
     TRENDBoard:WILLNUMBER←0
     TRENDBoard:CLOSENUMBER←0
     TRENDBoard:FARNUMBER←0
     TRENDBoard:WILLDETECT←NIL
     TRENDBoard:CLOSEDETECT←NIL
     TRENDBoard:FARDETECT←NIL
     LSTYPE←MATCHBoard:LAUNCHERTYPE;

IF (CAR TLIST) = 'NIL
THEN TRENDBoard:WILLDETECT←TGTWILLDETECT
     TRENDBoard:CLOSEDETECT←TGTCLOSEDETECT
     TRENDBoard:FARDETECT←TGTFARDETECT
(PRINTOUT T "This is WILLDETECT: " TGTWILLDETECT T
    "This is CLOSEDETECT: " TGTCLOSEDETECT T
    "THIS  is FARDETECT: " TGTFARDETECT T
    "This is WILLNUMBER: " TRENDBoard:WILLNUMBER T
    "this is CLOSENUMBER: " TRENDBoard:CLOSENUMBER T
   "This is FARNUMBER: " TRENDBoard:FARNUMBER T)
 (STOP);

THEN TPOINT←(CAR TLIST)
     TARGET←(CAR TPOINT)
     TRANGE←(CADDR TPOINT)
     THEADING←(CADR TPOINT)
     MARK←'T RANGEMARK←'T
     THEADIFF←(ABS (DIFFERENCE THEADING HEADING));

IF LSTYPE='TYPE1
THEN MAXRANGE←1325;

IF LSTYPE = 'TYPE2
THEN MAXRANGE←2650;

THEN RANGEDIFF←(DIFFERENCE MAXRANGE TRANGE);

     .
```

```
THEN RANGEDIFF←(DIFFERENCE MAXRANGE TRANGE);

IF RANGEDIFF < 0
THEN RANGEMARK←'NIL;

IF THEADIFF <= 1
   RANGEMARK = 'T
THEN TGTWILLDETECT←(APPEND (LIST TARGET) TGTWILLDETECT)   M
ARK←'NIL
 TRENDBoard:WILLNUMBER←(PLUS TRENDBoard:WILLNUMBER 1);

IF THEADIFF <= 5
   MARK = 'T
   RANGEMARK = 'T
THEN TGTCLOSEDETECT←(APPEND (LIST TARGET) TGTCLOSEDETECT) T
RENDBoard:CLOSENUMBER←(PLUS TRENDBoard:CLOSENUMBER 1)   MAR
K←'NIL;

IF THEADIFF <= 10
   MARK = 'T
   RANGEMARK = 'T
THEN TGTFARDETECT←(APPEND (LIST TARGET) TGTFARDETECT)
  TRENDBoard:FARNUMBER←(PLUS TRENDBoard:FARNUMBER 1);

THEN TLIST←(CDR TLIST);
#&(RuleSetSource "G↑U0.zI[:.n59.75")
29←
NIL
29←
NIL
29←
NIL
29←
NIL
29←
NIL
29←
NIL
29←
NIL
29←
NIL
29←
NIL
29←
NIL
29←
NIL
29←(REDO 1)
```

```
(TIMECHECK
  [LAMBDA NIL                                  (* edited: "12-Aug-85 16:17")
    (PROG (MATCHPOINT MATCHLIST READ)

          (* TIMECHECK takes the EVENTLIST and checks each event one by
one; if the ADS1 event has not been matched to the ADS2 event; it checks to see
if the PRESENTIME (which is taken off the most recent ADS1, ADS2 or BSS report
) exceeds the predicted time of arrival of the ADS2 report by more than 10 minute
s, For each event that fullfils this criterion; TIMECHECK asks the user if he wants it
 deleted from the eventlist or to keep it active.)


          (SETQ MATCHLIST EVENTLIST)
                                               (* If EVENTLIST is NIL, TIMECHE
CK just returns)
      LOOP(COND
            ((NULL (CAR MATCHLIST))
              (RETURN)))
          (SETQ MATCHPOINT (CAR MATCHLIST))
          (SETQ MATCHTIME (GETPROP MATCHPOINT
                                   (QUOTE P1TIME)))
          [COND
            ((NOT (NULL MATCHTIME))
              (COND
                ((GREATERP (ABS (DIFFERENCE PRESENTIME
                                            MATCHTIME))
                           (QUOTE 1000))
                                               (* In the SIMCOPE simulation 10
00 represents 10 minutes and 0 seconds)
                  (printout T T
                            "PRESENT TIME is currently: "
                              PRESENTIME T
                              "The ADS2 Report for: "
                              MATCHPOINT " was due at "
                              MATCHTIME " hours." T T
"Since the ADS2 Report is overdue by
more than 10 minutes,"
                              T
                         "SENTINEL would like to delete "
                              MATCHPOINT
                       " from the list of active Events."
                              T T
"If you concur type T, otherwise type NIL followed by a car
riage return to keep "
                              MATCHPOINT " active." T)
                  (SETQ READ (RATOM NIL NIL))
                  (COND
                    ((NULL READ)
                      (RETURN))
                    (T (SETQ EVENTLIST (REMOVE MATCHPOINT
                                               EVENTLIST))
```

```
                    (SETQ READ (RATOM NIL NIL))
                    (COND
                      ((NULL READ)
                        (RETURN))
                      (T (SETQ EVENTLIST (REMOVE MATCHPOINT
                                                 EVENTLIST))
```

(* If TIMECHECK is invoked from EVENTCHECK due to a spurious A DS2 report, the matchpoint must be removed from the TESTLIST so that the user may not be able to match an eliminated event,)

```
                         (COND
                           ((NOT (NULL TESTLIST))
                             (SETQ TESTLIST
                               (REMOVE MATCHPOINT TESTLIST]
          (SETQ MATCHLIST (CDR MATCHLIST))
          (GO LOOP])
(TIMECHECK)
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←
NIL
85←(REDO 100)
```

using $TESTHYPO
RuleSet TESTHYPO

```
Last Edited: 10-Oct-85 17:45:51 by
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:    TNUMBER RNUMBER MARK;
Control Structure: DOALL;
********


(* TEST is a hypothesis rule set that provides a "textbook"
 example of what a TEST event would result in.  This exampl
e can then be compared to the feature set of the event in q
uestion and a hypothesis selected.)

THEN TNUMBER←3 RNUMBER←0 MARK←'NIL;

(* TNUMBER records the number of factors that contribute to
 THREAT, RNUMBER number of rules triggered. TNUMBER  is set
 to 3 at outset and counted down.)
IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'TYPE1
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'TYPE2
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GetValue IntelBoard 'INTELCONDITION) = 'TEST
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'HIGH
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'EXTHIGH
THEN RNUMBER←(PLUS RNUMBER 1);

IF MATCHBoard:LAUNCHSITE ~= 'UNKNOWN
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1SENSOR) >= 7
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2SENSOR) >= 7
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN RNUMBER←(PLUS RNUMBER 1);
```

```
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN RNUMBER←(PLUS RNUMBER 1);

IF MATCHBoard:EXPNUMBER >= 3
THEN RNUMBER←(PLUS RNUMBER 1)
     TNUMBER←(SUB1 TNUMBER);

IF DetectBoard:EXPNUMBER < 3
THEN RNUMBER←(PLUS RNUMBER 2)
     TNUMBER←(SUB1 TNUMBER);

IF TRENDBoard:EXPNUMBER >= 3
THEN RNUMBER←(PLUS RNUMBER 1);

IF TRENDBoard:HIVALNUMBER <= 2
THEN RNUMBER←(PLUS RNUMBER 2)
     TNUMBER←(SUB1 TNUMBER);

THEN (SETQ NOTHREATNUM TNUMBER)
     (SETQ TESTNUM RNUMBER);
#&(RuleSetSource "SPU0.zI[:.;d9.7")
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←
NIL
91←(REDO 100)
```

using $TrendCheck

RuleSet TrendCheck

Last Edited: 3-Oct-85 18:43:18 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   FIRSTCONF SECONDCONF UPNUMBER MARK;
Control Structure: DOALL;
********



(* TrendCheck weighs the evidence from 3 factors CONFIDENCE
 LEVELS, IR TYPE and SENSOR RATING from ADS-1 to      the A
DS-2 Report and attempts to discern whether the    Trend of
 evidence is going up or down.)

THEN UPNUMBER←0 ;

(* To make it easier to compare, first P1CONFIDENCE and P2C
ONFIDENCE from the EVENT LIST are mapped to numbers.)

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'EXTHIGH
THEN FIRSTCONF←5;

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'HIGH
THEN FIRSTCONF←4;

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'MIXED
THEN FIRSTCONF←3;

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'LOW
THEN FIRSTCONF←2;

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'EXTLOW
THEN FIRSTCONF←1;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTHIGH
THEN SECONDCONF←5;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'MIXED
THEN SECONDCONF←3;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'HIGH
THEN SECONDCONF←4;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN SECONDCONF←2;

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'EXTLOW
THEN SECONDCONF←1;
```

```
(* Here they are compared, in case of tie, TREND is UP.)

IF SECONDCONF >= FIRSTCONF
THEN UPNUMBER←(PLUS UPNUMBER 1);

(* The Infrared Type TREND takes advantage of the typing of
 ADS-2 Feature Set into MATCH, MISMATCH, NOMATCH. Only way
to get DOWN trend is to get MISMATCH or a MATCH on an Indis
tinct or Unknown ADS-1 reading.)

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'INDISTINCT
   (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN UPNUMBER←(SUB1 UPNUMBER);

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'UNKNOWN
   (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN UPNUMBER←(SUB1 UPNUMBER);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'MATCH
THEN UPNUMBER←(PLUS UPNUMBER 1);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'NOMATCH
THEN UPNUMBER←(PLUS UPNUMBER 1);

(* SENSOR Ratings can be compared directly since they
are already numbers. TIE is an UP vote.)
IF (GETPROP EVENTMATCH 'P2SENSOR) >= (GETPROP EVENTMATCH 'P
1SENSOR)
THEN UPNUMBER←(PLUS UPNUMBER 1);

 (* Now the results are interpreted, COMMENT and EXPNUMBER
on the TRENDBoard are set.)

 IF UPNUMBER = 3
 THEN TRENDBoard:EXPNUMBER←4
      TRENDBoard:COMMENT←'(PRINTOUT T "Trend of Evidence is
: POSITIVELY UP" T);

 IF UPNUMBER = 2
THEN TRENDBoard:EXPNUMBER←3
    TRENDBoard:COMMENT←'(PRINTOUT T "Trend of Evidence is:
UP " T);

 IF UPNUMBER = 1
THEN TRENDBoard:EXPNUMBER←2
    TRENDBoard:COMMENT←'(PRINTOUT T "Trend of Evidence is:
DOWN " T);
 IF UPNUMBER = 0
THEN•TRENDBoard:EXPNUMBER←1
    TRENDBoard:COMMENT←'(PRINTOUT T "Trend of Evidence is:
POSITIVELY DOWN" T);
```

using $UNKNOWN

RuleSet UNKNOWN

Last Edited: 3-Oct-85 18:35:58 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   TNUMBER RNUMBER MARK;
Control Structure: DOALL;
********


(* UNKNOWN is a hypothesis rule set that provides a      "t
extbook" example of what an UNKNOWN  event is. This   examp
le can then be compared to the feature set of the event in
question and a hypothesis selected.)

THEN TNUMBER←3 RNUMBER←0 MARK←'NIL;

(* TNUMBER records the number of factors that contribute to
 THREAT, RNUMBER number of rules triggered. TNUMBER   is se
t to 3 and then counted down when rules say so.)

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'INDISTINCT
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'UNKNOWN
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GetValue IntelBoard 'INTELCONDITION) = 'GREEN
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'LOW
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'EXTLOW
THEN RNUMBER←(PLUS RNUMBER 1);

IF MATCHBoard:LAUNCHSITE = 'UNKNOWN
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1SENSOR) <= 5
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'MISMATCH
THEN RNUMBER←(PLUS RNUMBER 1);

(* SensorMap takes the P2SENSOR rating and returns it toits
 constituent parts, Sensor Confidence and Capability)
THEN (SETQ SENSORNUM (GETPROP EVENTMATCH 'P2SENSOR))
     (RunRS 'SensorMap);
```

```
THEN RNUMBER←(PLUS RNUMBER 1);

(* SensorMap takes the P2SENSOR rating and returns it toits
 constituent parts, Sensor Confidence and Capability)
THEN (SETQ SENSORNUM (GETPROP EVENTMATCH 'P2SENSOR))
     (RunRS 'SensorMap);

IF SENSORCONF ~= 'HIGH
THEN RNUMBER←(PLUS RNUMBER 1);

IF SENSORCAP ~= 'FULL
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) =  'EXTLOW
THEN RNUMBER←(PLUS RNUMBER 1);

IF MATCHBoard:EXPNUMBER <= 2
THEN RNUMBER←(PLUS RNUMBER 1);

(* When Launchsite heading is unknown it appears as  a    "
XXX" on the ADS-2 message.)

IF MATCHBoard:LSHEADING = 'XXX
THEN RNUMBER←(PLUS RNUMBER 2);

IF TRENDBoard:EXPNUMBER <= 2
THEN RNUMBER←(PLUS RNUMBER 2);

THEN (SETQ UNKNOWNNUM RNUMBER);
#&(RuleSetSource "SPU0.zI[:.;d9.9")
96←
NIL
96←
NIL
96←
NIL
96←
NIL
96←
NIL
96←
NIL
96←
NIL
96←
NIL
96←(REDO 100)
```

```
using $UNRESOLVED
```

RuleSet UNRESOLVED

Last Edited: 3-Oct-85 18:37:08 by

```
WorkSpace Class: Boards;
Compiler Options:  ;
Temporary Vars:   TNUMBER RNUMBER MARK;
Control Structure: DOALL;
********



(* UNRESOLVED is a hypothesis rule set that provides a "tex
tbook" example of what an UNRESOLVED event is. This   examp
le can then be compared to the feature set of the event in
question and a hypothesis selected.)

THEN TNUMBER←3 RNUMBER←0 MARK←'NIL;

(* TNUMBER records the number of factors that contribute to
 THREAT, RNUMBER number of rules triggered. TNUMBER   is se
t to 3 and then counted down when rules say so.)

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'TYPE1
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1IRTYPE) = 'TYPE2
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'LOW
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1CONFIDENCE) = 'MIXED
THEN RNUMBER←(PLUS RNUMBER 1);

IF MATCHBoard:LAUNCHSITE ~= 'UNKNOWN
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P1SENSOR) < 7
   (GETPROP EVENTMATCH 'P1SENSOR) >= 4
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2TYPE) = 'NOMATCH
THEN RNUMBER←(PLUS RNUMBER 1);

(* P2SENSOR is broken down into its constituent parts:    a
 SENSOR Confidence and a Capability.)

THEN (SETQ SENSORNUM (GETPROP EVENTMATCH 'P2SENSOR))
     (RunRS 'SensorMap);

     •
```

```
THEN (SETQ SENSORNUM (GETPROP EVENTMATCH 'P2SENSOR))
     (RunRS 'SensorMap);

IF SENSORCONF = 'MEDUIM
THEN RNUMBER←(PLUS RNUMBER 1);

IF SENSORCAP = 'PARTIAL
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) = 'LOW
THEN RNUMBER←(PLUS RNUMBER 1);

IF (GETPROP EVENTMATCH 'P2CONFIDENCE) =  'MIXED
THEN RNUMBER←(PLUS RNUMBER 1);

IF MATCHBoard:EXPNUMBER <= 2
THEN RNUMBER←(PLUS RNUMBER 1)
     TNUMBER←(SUB1 TNUMBER);

IF DetectBoard:EXPNUMBER <= 3
   DetectBoard:EXPNUMBER >= 2
THEN RNUMBER←(PLUS RNUMBER 1)
     TNUMBER←(SUB1 TNUMBER);

IF TRENDBoard:EXPNUMBER <= 3
   TRENDBoard:EXPNUMBER >= 2
THEN RNUMBER←(PLUS RNUMBER 2);

IF TRENDBoard:HIVALNUMBER <= 2
THEN RNUMBER←(PLUS RNUMBER 2)
     TNUMBER←(SUB1 TNUMBER);

THEN (SETQ NOTHREATNUM TNUMBER)
     (SETQ UNRESOLVEDNUM RNUMBER);
#&(RuleSetSource "SPU0.zI[:.;d9.10")
1←
NIL
1←
NIL
1←
NIL
1←
NIL
1←
NIL
1←
NIL
1←
NIL
1←(RS232.PRINT (WHICHW) T)
```

VITA

Daniel Louis Tobat was born on 24 August, 1959 in Torrejon Air Base, Spain. He graduated from the Franklin D. Roosevelt American High School in Lima, Peru in June 1977. In May 1981, he graduated from the United States Air Force Academy with a Bachelor of Science in Electrical Engineering and was commissioned as a second lieutenant. He then completed the six month long Communications-Electronics Officer School as a distinguished graduate at Keesler Air Force Base, Mississippi in December 1981. From January 1982 to May 1984, he served as a communications-electronics engineer with the 1836 Engineering and Installation Group at Lindsey Air Station, Wiesbaden, West Germany. In May 1984, he was selected to pursue a Master's of Science degree in Electrical Engineering at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. Following graduation in December 1985, Captain Tobat is being assigned to the 2045 Information Systems Group, Andrews Air Force Base, Maryland.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A163 929

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/85D-50	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering — 6b. OFFICE SYMBOL (If applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) See Box 19	

12. PERSONAL AUTHOR(S)
Daniel L. Tobat, B.S.E.E., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM ______ TO ______	1985 December	334

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Artificial Intelligence , Missile Warning System
09	02		Expert System , Decision Aid
06	04		Command, Control and Communications

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: An Expert System Decision Aid for a Command, Control and Communications Operator.

Thesis Chairman: Steven K. Rogers, Captain USAF

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 16 JAN 86
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Steven K. Rogers, Captain, USAF	(513) 255-3376	AFIT/ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

The growing complexity and quantity of information used in Command, Control and Communications (C3) networks makes it essential to reduce the workload on the operators of these networks. The SENTINEL project uses the Artificial Intelligence concept of an expert system to produce a decision aid for the strategic Missile Warning Officer, using a simulation of a C3 network that involves multiple missile launches and up to 20 countries. In this research, a blackboard model expert system using rule bases and object oriented programming techniques was developed to permit SENTINEL to offer several layers of explanation. SENTINEL analyzes patterns and causes of reported events into higher level yet less precise forms to provide the upper layer of explanation. SENTINEL deals with uncertainty by using the statistical concepts of feature sets and decision thresholding. The feature sets represent the essential characteristics of a launch event and are evaluated to see how well they fit a particular hypothesis. The decision threshold used to select an interpretation is determined by appraising the distance from each hypothesis, as well as by previous events. This project demonstrates the feasibility of building expert system decision aids for C3 operators by using specialized explanation capabilities, and reasoning with uncertainty in a more statistically conventional way. (THESIS).